SAMA7G5 Series

Up to 1-GHz Arm® Cortex®-A7, MIPI Camera, Dual Ethernet, Audio and Security

Introduction

The SAMA7G5 is a high-performance, ultra-low power Arm Cortex-A7 CPU-based embedded microprocessor (MPU) running up to 1 GHz, with support for multiple memories such as 16-bit DDR2, DDR3, DDR3L, LPDDR2, LPDDR3, octal/quad SPI and e.MMC Flash.

The SAMA7G5 integrates complete imaging and audio subsystems with 12-bit parallel and/or MIPI-CSI2 camera interfaces up to 8 Mpixels, up to four I^2S, one SPDIF transmitter and receiver and a 4-stereo channel audio sample rate converter.

The device also features a large number of connectivity options with Dual Ethernet (one Gigabit Ethernet and one 10/100 Ethernet), six CAN-FD and three high-speed USB and offers advanced security functions (secure boot, secure key storage, high-performance crypto accelerators for AES, SHA, RSA and ECC).

Features

- Arm Cortex-A7 Core
  - Arm TrustZone®
  - NEON multimedia architecture
  - Floating Point Unit
  - Embedded Trace module with instruction trace stream, including 16 Kbytes of CoreSight™ Embedded Trace buffer
  - 32 Kbytes of L1 data cache, 32 Kbytes of L1 instruction cache
  - 256 Kbytes of L2 cache
  - Up to 1 GHz operational frequency
  - Voltage and frequency scaling support
  - 64-bit generic timers
- Internal Memory Architecture
  - 128 Kbytes of internal SRAM and 5 Kbytes of secure backup RAM
  - 80 Kbytes of maskable ROM, embedding a secure boot loader (boot on QSPI NOR, SLC NAND, SD, e.MMC)
  - 96 Kbytes ROM for NAND Flash ECC tables
  - 40 Kbytes ROM for crypto-libraries (RSA, ECC, etc.)
  - 11 Kbytes of internal OTP
- External Memory Support
  - 16-bit high-bandwidth, double data rate multi-port dynamic RAM controller. Supports up to 16 Gbit 8-bank DDR2/DDR3/DDR3L/LPDDR2/LPDDR3 up to 533 MHz
  - 16-bit static memory controller, FPGA support with synchronous clock
  - 8-bit SLC and MLC NAND controller with up to 32-bit error correcting code
  - One 8-bit high-speed memory card host e.MMC 5.1 (HS400), SD3.0 SDR104 mode support
  - Two 4-bit high-speed memory card hosts e.MMC 4.51 (HS200), SD3.01 SDR104 mode support
  - One octal Serial Peripheral Interface running up to 200 MHz DDR

  - One quad Serial Peripheral Interface
- System
  - Power-on reset cells, reset controller, shutdown controller, watchdog and secure watchdog timers running on internal slow RC oscillator (32 kHz typical) and real-time clock running on slow crystal oscillator (32.768 kHz)
  - Two internal trimmed RC oscillators with typical values: 32 kHz and 12 MHz
  - Two crystal oscillators: 32.768 kHz and 20 to 50 MHz
  - Eight PLLs for core, system bus and peripherals, serial interfaces, DDR I/Os, pixel clock, audio, USB, MIPI CSI-2 and Ethernet
  - Two 32-channel DMA with per-channel security configuration
  - One 8-channel DMA dedicated to memory-to-memory transactions
  - Eight programmable clock output signals
- Power Considerations
  - Different power domains and power modes to reduce power consumption
  - Low-power consumption in Backup mode with 5 Kbytes of secure backup SRAM and DDR-SDRAM in Self-Refresh mode
  - Low-power with SRAM and register retention, wake-up from various events (USB, CAN, Ethernet WOL, FLEXCOMs), internal events (RTC, timer) and I/O activity
  - Embedded LDOs for MIPI CSI-2, analog and PLLs, to enable low-cost power management solutions
  - Optimum connection to Microchip MCP16501/2 PMICs to enter and exit various power modes of the application.
- Multimedia Peripherals
  - Audio
    - Two synchronous serial controllers, each with 16 channels of up to 32-bit TDM data
    - One inter-IC sound multi-channel controller with TDM256 support
    - Up to two 4-channel pulse density microphone controllers; support for eight microphones in parallel
    - One Sony/Philips digital interface transmitter and receiver
    - Audio sample rate converter including four stereo channels
  - Image
    - Image sensor controller, ITU-R BT. 601/656 supporting up to eight megapixels for still images and 60 fps in 720p mode, 8 bits, raw Bayer, YCbCr, monochrome, camera ISP
    - 2-lane MIPI CSI-2 (D-PHY) and 12-bit RGB interface support
- Peripherals
  - Two high-speed USB devices and three high-speed USB hosts sharing three on-chip transceivers
  - One 10/100/1000 Gigabit Ethernet MAC supporting RGMII, MII and RMII (GMAC0) and one 10/100 Ethernet MAC supporting MII and RMII (GMAC1) compliant with:
    - IEEE802.3az Energy-Efficient Ethernet
    - IEEE802.1AS Timestamping for Ethernet AVB support
    - IEEE802.1Qav Credit-based traffic shaping hardware support
    - IEEE1588 Precision Time Protocol
    - IEEE1588 Timestamp Unit (TSU) with TSU timer comparison signal triggering a timer counter and available on a PIO line
  - Six flexible data rate CAN-FD controllers with SRAM-based mailboxes with time- and event-triggered transmission
  - Twelve FLEXCOMs (USART, SPI and TWIHS)
  - Six 64-bit timers
  - Two three-channel 32-bit timer counters, with PWM generation
  - One four-channel 16-bit PWM controller
  - One 16-channel 12-bit analog-to-digital converter, up to 1 Msps
- Safety
  - Temperature and core voltage monitoring

- Zero-power power-on reset cells
- Main crystal monitor and clock failure detector with failsafe switchover to main RC oscillator
- 32 kHz crystal monitor and clock failure detector with failsafe switchover to internal 32 kHz RC oscillator
- Integrity check monitor based on SHA256
- Safety critical modules (WDT, RSTC, SHDWC, etc.) running on always-on slow RC oscillator
- Register write protection

- Security
  - TrustZone support
  - One Secure TrustZone watchdog timer running on RC oscillator, providing protection against TrustZone starvation
  - Temperature, voltage and frequency monitoring
  - Secure backup SRAM
    - 5 Kbytes scrambled with non-imprinting support powered with VBAT or VDDIN33:
      - 1 Kbyte non erasable on tamper detection
      - 4 Kbytes erasable on tamper detection
  - Four tamper pins for static or dynamic detection
    - Can be used as regular wake-up lines
  - 256-bit general purpose backup register, erasable on tamper detection
  - Programmable OTP with bits available for user purposes
  - Configurable JTAG security (full debug, non-secure-only debug, no debug)
  - 128-bit AES on-the-fly encryption/decryption on DDR memory, SMC, QSPI0 and QSPI1, including automatic key load at start-up. Separate keys for secure and non-secure accesses (TZAESB).
  - True random number generator compliant with NIST Special Publication 800-22 Tests Suite and FIPS PUB 140-2 and 140-3
  - Secure RTC
- Cryptography
  - SHA (SHA1, SHA224, SHA256, SHA384, SHA512) compliant with FIPS Publications 180-2
  - AES: 256-, 192-, 128-bit key algorithm, compliant with FIPS PUB 197 specifications
  - TDES: two-key or three-key algorithms, compliant with FIPS PUB 46-3 specifications
  - Public Key Coprocessor (CPKCC) and associated Classical Public Key Cryptography Library (CPKCL) for RSA, DSA, ECC GF(2^n), ECC $GF_{(p)}$[(1)]
- Up to 136 I/Os
  - Fully programmable through set/clear registers
  - Multiplexing of eight peripheral functions per I/O line
  - Each I/O line can be assigned to a peripheral or used as a general purpose I/O
  - Synchronous output, possibility to set or clear simultaneously up to 32 I/O lines in a single write
  - General purpose analog and digital inputs tolerant to positive and negative current injection
- Design for Low ElectroMagnetic Interference (EMI)
  - Slewrate-controlled I/Os
  - DDR Phy with impedance-calibrated drivers
  - Spread spectrum PLLs
  - Careful BGA power/ground ball assignment to provide optimum decoupling capacitors placement
- Microchip Recommended Power Management Integrated Circuits (PMICs)
  - MCP16502, 6-channel PMIC with I²C control interface; supports dynamic voltage scaling and processor Low-Power modes (ULP2, BSR)
  - MCP16501, 4-channel PMIC optimized for compact PCB layout; supports processor Low-Power mode (BSR)
- Junction Temperature (T_J) Range
  - Industrial : -40°C to +105°C
- Package

– 14x14 mm^2, 0.65 mm pitch, 343-ball TFBGA, optimized for standard class PCB layout (down to four layers)

Note:

1. CPKCC and CPKCL are described in the application note "Using CPKCL Version 02.08.01.xx on SAMA7G5" (document no. DS00004438). This document is available under Non-Disclosure Agreement (NDA). Contact a Microchip Sales Representative for further details.

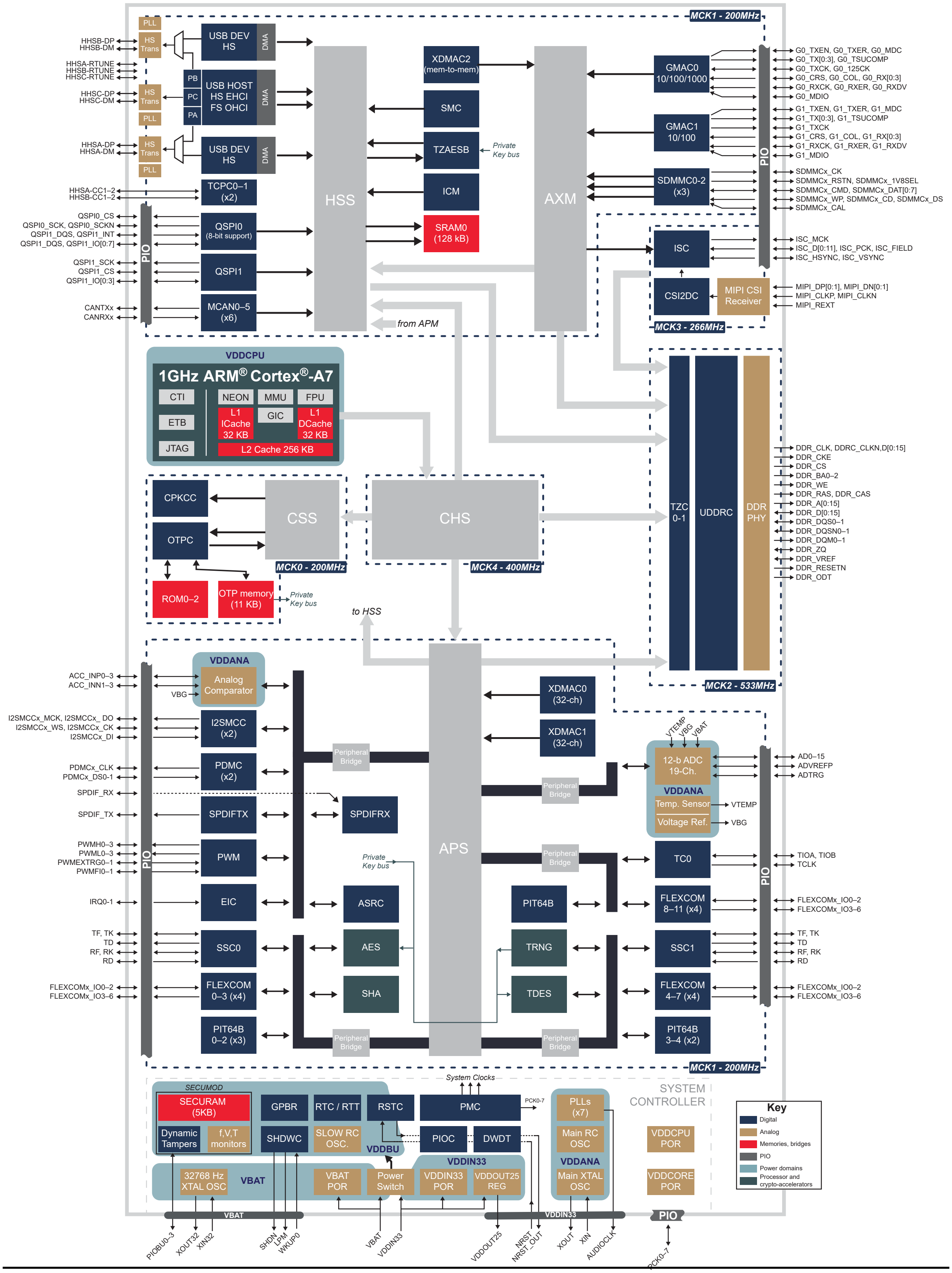
1. Configuration Summary

Table 1-1. SAMA7G5 Configuration Summary

Feature	SAMA7G54
Package	BGA343
External memory support	NAND Flash, LPDDR2, LPDDR3, DDR2,3(L), QSPI (NAND, NOR)
Number of PIOs	136
SDMMC	3
DDR bus	16-bit
GMAC	RGMII / MII/RMII + MII / RMII
CAN-FD	6
FLEXCOM	12
ADC inputs	16
USB device/host	2/2 sharing 2 USB Type-C™ transceivers + 1 host
I2SMCC channels output/input	8/8
SSC	2
ISC	MIPI CSI-2 + Parallel RGB
PDMC channels (2 microphones per channel)	Up to 4

2. Block Diagram

Figure 2-1. SAMA7G5 Series Block Diagram



3. Signal Description

Table 3-1. Signal Description List

Signal Name	Function	Type	Comments	Active Level
Clocks, Oscillators and PLLs				
XIN	Main Oscillator Input	Input	–	–
XOUT	Main Oscillator Output	Output	–	–
XIN32	Slow Clock Oscillator Input	Input	–	–
XOUT32	Slow Clock Oscillator Output	Output	–	–
AUDIOCLK	Audio Clock	Output	–	–
PCK[0..7]	Programmable Clock Output	Output	Reset State: - PIO Input - Internal Pull-up enabled - Schmitt Trigger enabled	–
Shutdown, Wake-up Logic				
LPM	Low-power Mode	Output	–	–
SHDN	Shutdown Control	Output	–	–
WKUP[0..5]	Wakeup Input	Input	–	–
ICE and JTAG				
TCK/SWCLK	Test Clock/Serial Wire Clock	Input	–	–
TDI	Test Data In	Input	–	–
TDO	Test Data Out	Output	–	–
TMS/SWDIO	Test Mode Select/Serial Wire Input/Output	I/O	–	–
JTAGSEL	JTAG Selection	Input	–	–
Reset/Test				
NRST	Microprocessor Reset	Input	–	Low
TST	Test Mode Select	Input	–	–
NTRST	Test Reset Signal	Input	–	–
NRST_OUT	Microprocessor Reset Output	Output	–	Low
External Interrupt Controller - EIC				
IRQ[0..1]	External Interrupt Input	Input	–	–
PIO Controller				
PA0–PAxx	Parallel IO Controller	I/O	–	–
PB0–PBxx	Parallel IO Controller	I/O	–	–
PC0–PCxx	Parallel IO Controller	I/O	–	–
PD0–PDxx	Parallel IO Controller	I/O	–	–
PE0–PExx	Parallel IO Controller	I/O	–	–

...........continued				
Signal Name	Function	Type	Comments	Active Level
		External Bus Interface - EBI		
D[15..0]	Data Bus	I/O	–	–
A[25..0]	Address Bus	Output	–	–
NWAIT	External Wait Signal	Input	–	Low
		Static Memory Controller - SMC		
NCS[0..3]	Chip Select Lines	Output	–	Low
NWR[0..1]	Write Signal	Output	–	Low
NRD	Read Signal	Output	–	Low
NWE	Write Enable	Output	–	Low
NBS[0..1]	Byte Mask Signal	Output	–	Low
NANDOE	NAND Flash Output Enable	Output	–	Low
NANDWE	NAND Flash Write Enable	Output	–	Low
		DDR2/DDR3(L)/LPDDR2/LPDDR3 Controller		
DDR_CLK, DDR_CLKN	DDR Differential Clock	Output	–	–
DDR_CKE	DDR Clock Enable	Output	–	High
DDR_CS	DDR Controller Chip Select	Output	–	Low
DDR_BA[2..0]	Bank Select	Output	–	Low
DDR_WE	DDR Write Enable	Output	–	Low
DDR_RAS, DDR_CAS	Row and Column Signal	Output	–	Low
DDR_A[15..0]	DDR Address Bus	Output	–	–
DDR_D[15..0]	DDR Data Bus	I/O	–	–
DDR_DQS[1..0], DDR_DQSN[1..0]	Differential Data Strobe	I/O	–	–
DDR_DQM[1..0]	Write Data Mask	Output	–	–
DDR_ZQ	DDR/LPDDR Calibration	Input	–	–
DDR_VREF	DDR/LPDDR Reference	Input	–	–
DDR_RESETN	DDR3 Active Low Asynchronous Reset	Output	–	Low
DDR_ODT	DDR3 On-Die Termination	Output	–	High
		Secure Data Memory Card - SDMMCx [2..0]		
SDMMCx_CAL	SD Card Calibration	Input	–	Low
SDMMCx_CD	SD Card / e.MMC Card Detect	Input	–	Low
SDMMCx_CMD	SD Card / e.MMC Command line	I/O	–	–
SDMMCx_WP	SD Card Connector Write Protect Signal	Input	–	High
SDMMCx_RSTN	e.MMC Reset Signal	Output	–	Low
SDMMCx_1V8SEL	SD Card Signal Voltage Selection	Output	–	–

...........continued				
Signal Name	**Function**	**Type**	**Comments**	**Active Level**
SDMMCx_CK	SD Card / e.MMC Clock Signal	Output	–	–
SDMMCx_DAT[3..0]	SD Card	I/O	–	–
SDMMC0_DAT[7..4]	e.MMC Data Lines	I/O	–	–
SDMMC0_DS	e.MMC Data Strobe	Input	–	–
Flexible Serial Communication Controller - FLEXCOMx [11..0]				
FLEXCOMx_IO0	FLEXCOMx Transmit Data	I/O	–	–
FLEXCOMx_IO1	FLEXCOMx Receive Data	I/O	–	–
FLEXCOMx_IO2	FLEXCOMx Serial Clock	I/O	–	–
FLEXCOMx_IO3	FLEXCOMx Clear To Send / Peripheral Chip Select	I/O	–	–
FLEXCOMx_IO4	FLEXCOMx Request To Send / Peripheral Chip Select	Output	–	–
FLEXCOMx_IO5	FLEXCOMx SPI Chip Select 2	Output	–	–
FLEXCOMx_IO6	FLEXCOMx SPI Chip Select 3	Output	–	–
Inter-IC Sound Multi Channel Controller - I2SMCCx [1..0]				
I2SMCCx_MCK	I^2S Bus Clock	Output	–	–
I2SMCCx_CK	Serial Clock	I/O	–	–
I2SMCCx_WS	I^2S Word Select	I/O	–	–
I2SMCCx_DIN[0..3]	Serial Data Inputs	Input	–	–
I2SMCCx_DOUT[0..3]	Serial Data Outputs	Output	–	–
Synchronous Serial Controller - SSCx [1..0]				
TDx	SSC Transmit Data	Output	–	–
RDx	SSC Receive Data	Input	–	–
TKx	SSC Transmit Clock	I/O	–	–
RKx	SSC Receive Clock	I/O	–	–
TFx	SSC Transmit Frame Sync	I/O	–	–
RFx	SSC Receive Frame Sync	I/O	–	–
Timer/Counter - TCx [1..0]				
TCLK[5..0]	TC Channel y External Clock Input	Input	–	–
TIOA[5..0]	TC Channel y I/O Line A	I/O	–	–
TIOB[5..0]	TC Channel y I/O Line B	I/O	–	–
Quad IO SPI - QSPIx [1..0]				
QSPIx_SCK	QSPI Serial Clock	Output	–	–
QSPIx_CS	QSPI Chip Select	Output	–	Low

...........continued				
Signal Name	**Function**	**Type**	**Comments**	**Active Level**
QSPIx_IO[0..3]	QSPI I/O QIO0 is QMOSI Host Out - Client In QIO1 is QMISO Host In - Client Out	I/O	–	–
QSPI0_IO[7..4]	QSPI0 I/Os for Octal Mode	I/O	–	–
QSPI0_SCKN	Negative QSPI0 Serial Clock	Output	–	–
QSPI0_INT	QSPI Interrupt	Input	–	Low
QSPI0_DQS	QSPI0 Data Strobe	Input	–	–
Pulse Width Modulation Controller - PWM				
PWMH[0..3]	PWM Waveform Output High	Output	–	–
PWML[0..3]	PWM Waveform Output Low	Output	–	–
PWMFI[0..1]	PWM Fault Inputs	Input	–	–
PWMEXTRG[0..1]	PWM External Trigger	Input	–	–
USB High Speed Ports - A, B, C				
HHSA-DP	USB Host Port A High Speed Data + USB Device A High Speed Data +	Analog	–	–
HHSA-DM	USB Host Port A High Speed Data - USB Device A High Speed Data -	Analog	–	–
HHSB-DP	USB Host Port B High Speed Data + USB Device B High Speed Data +	Analog	–	–
HHSB-DM	USB Host Port B High Speed Data - USB Device B High Speed Data -	Analog	–	–
HHSC-DP	USB Host Port C High Speed Data +	Analog	–	–
HHSC-DM	USB Host Port C High Speed Data -	Analog	–	–
HHSA-CC1-2	USB Host Port A Configuration Channel 1 and 2	Analog	–	–
HHSA-RTUNE	USB Host Port A Tune	Analog	–	–
HHSB-CC1-2	USB Host Port B Configuration Channel 1 and 2	Analog	–	–
HHSB-RTUNE	USB Host Port B Tune	Analog	–	–
HHSC-RTUNE	USB Host Port C Tune	Analog	–	–
Ethernet 10/100/1000 - GMAC0				
G0_TXCK/G0_REFCK	Transmit Clock or 50 MHz Reference Clock	I/O	–	–
G0_125CK	125 MHz Clock	I/O	–	–
G0_TXEN/G0_TXCTL	Transmit Enable or Transmit Control Signal	Output	–	–
G0_TX[3..0]	Transmit Data	Output	–	–
G0_TXER	Transmit Coding Error	Output	–	–
G0_RXCK	Receive Clock	Input	–	–

...........continued

Signal Name	Function	Type	Comments	Active Level
G0_RXDV/G0_CRSDV/ G0_RXCTL	Receive Data Valid or Carrier Sense and Data Valid or Receive Control Signal	Input	–	–
G0_RX[3..0]	Receive Data	Input	–	–
G0_RXER	Receive Error	Input	–	–
G0_CRS	Carrier Sense	Input	–	–
G0_COL	Collision Detect	Input	–	–
G0_MDC	Management Data Clock	Output	–	–
G0_MDIO	Management Data Input/Output	I/O	–	–
G0_TSUCOMP	TSU Timer Comparison Valid	Output	–	–
Ethernet 10/100 - GMAC1				
G1_TXCK/G1_REFCK	Transmit Clock or 50 MHz Reference Clock	I/O	–	–
G1_TXEN	Transmit Enable	Output	–	–
G1_TX[3..0]	Transmit Data	Output	–	–
G1_TXER	Transmit Coding Error	Output	–	–
G1_RXCK	Receive Clock	Input	–	–
G1_RXDV/G1_CRSDV	Receive Data Valid or Carrier Sense and Data Valid	Input	–	–
G1_RX[3..0]	Receive Data	Input	–	–
G1_RXER	Receive Error	Input	–	–
G1_CRS	Carrier Sense	Input	–	–
G1_COL	Collision Detect	Input	–	–
G1_MDC	Management Data Clock	Output	–	–
G1_MDIO	Management Data Input/Output	I/O	–	–
G1_TSUCOMP	TSU Timer Comparison Valid	Output	–	–
Analog-to-Digital Converter - ADC				
AD[0..15]	16 Analog Inputs	Analog	–	–
ADTRG	ADC Trigger	Input	–	–
ADVREFP	ADC Reference	Analog	–	–
Secure Box Module - SBM				
PIOBU[0..3]	Tamper I/Os	I/O	–	–
Image Sensor Controller - ISC				
ISC_D[0..11]	Image Sensor Data	Input	–	–
ISC_HSYNC	Image Sensor Horizontal Synchro	Input	–	–
ISC_VSYNC	Image Sensor Vertical Synchro	Input	–	–
ISC_PCK	Image Sensor Pixel clock	Input	–	–
ISC_MCK	Image Sensor Main clock	Output	–	–

...........continued				
Signal Name	**Function**	**Type**	**Comments**	**Active Level**
ISC_FIELD	Field Identification Signal	Input	–	–
Controller Area Network - CANx [5..0]				
CANRXx	CAN Receive	Input	–	–
CANTXx	CAN Transmit	Output	–	–
Pulse Density Modulation Interface Controller - PDMCx [1..0]				
PDMCx_DS[0..1]	PDM Data	Input	–	–
PDMCx_CLK	PDM Clock	Output	–	–
Sony Philips Digital Interface Receiver - SPDIFRX				
SPDIF_RX	Receive Data	Input	–	–
Sony Philips Digital Interface Transmitter - SPDIFTX				
SPDIF_TX	Transmit Data	Output	–	–
MIPI DPHY				
MIPI_DP[0..1]/ MIPI_DN[0..1]	MIPI DPHY Differential Input Data Lane 0..1	Input	–	–
MIPI_CLKP/MIPI_CLKN	MIPI DPHY Differential Input Clock Lane	Input	–	–
MIPI_REXT	Calibration Reference Resistor	Input	4.02 KΩ E96	–

4. Microchip Recommended Power Management Solutions

MCP16502 and MCP16501 are multi-channel Power Management Integrated Circuits (PMICs) designed for the SAMA7G5 series.

4.1 MCP16502 PMIC

MCP16502 is a 6-channel PMIC supporting high-performance applications requiring Digital Voltage Scaling.

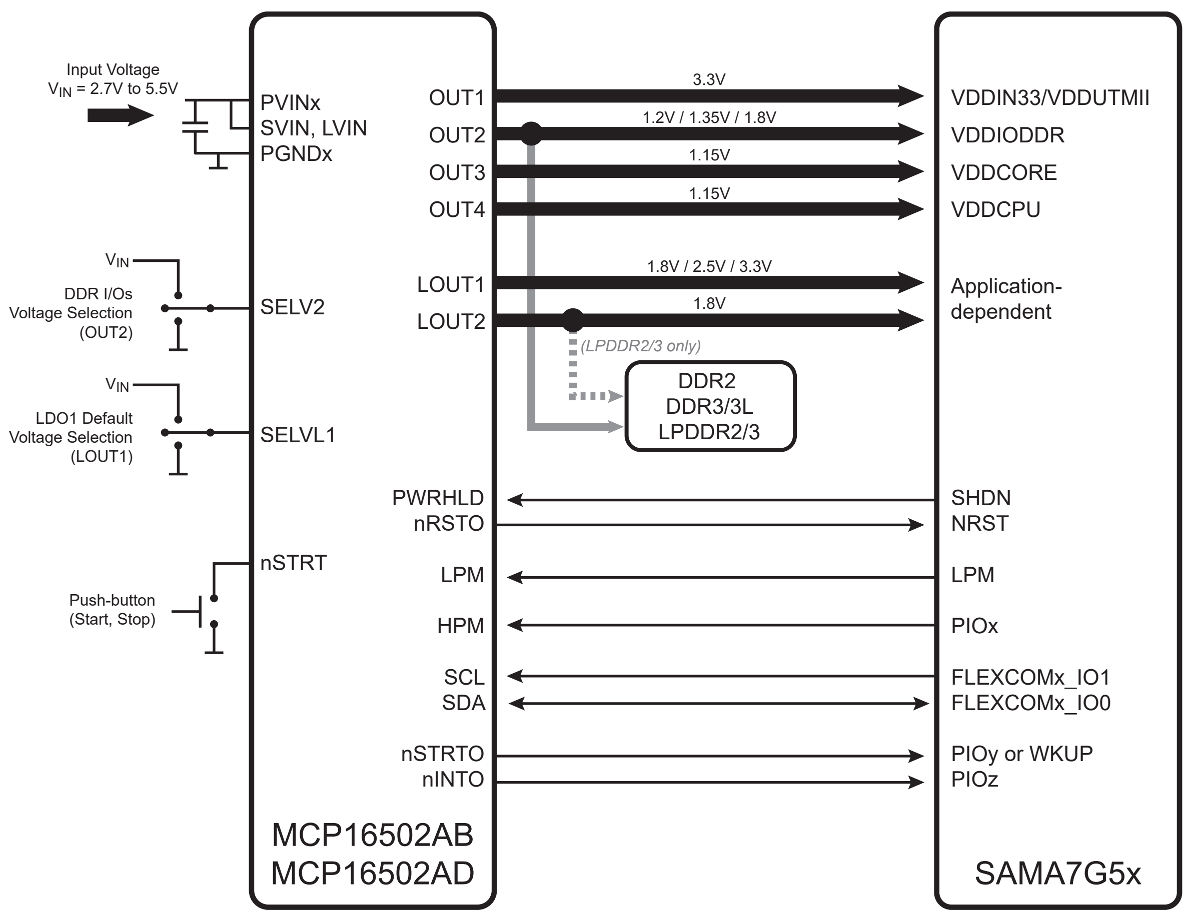
MCP16502 features four 1A DC-DC Buck regulators and two 0.3A auxiliary LDO regulators, and provides a comprehensive interface to the MPU, which includes an interrupt flag and a 1-MHz I²C interface.

The PMIC-processor interface is optimized so that it remains leakage-free in any power mode. MCP16502 supports all SAMA7G5x low-power modes. Although MCP16502 power states are fully programmable through its I²C interface, they are entered and/or exited on pin level change to ease software operations. See the following figure.

- MCP16502**AB** supports systems based on DDR2 (1.8V) or DDR3L (1.35V) SDRAM memories.
- MCP16502**AD** supports applications using LPDDR2-SDRAM or LPDDR3-SDRAM. In this version, LOUT2 is started at 1.8V.

For further details, refer to the MCP16502 documentation on www.microchip.com.

Figure 4-1. MCP16502 Simplified Application Block Diagram



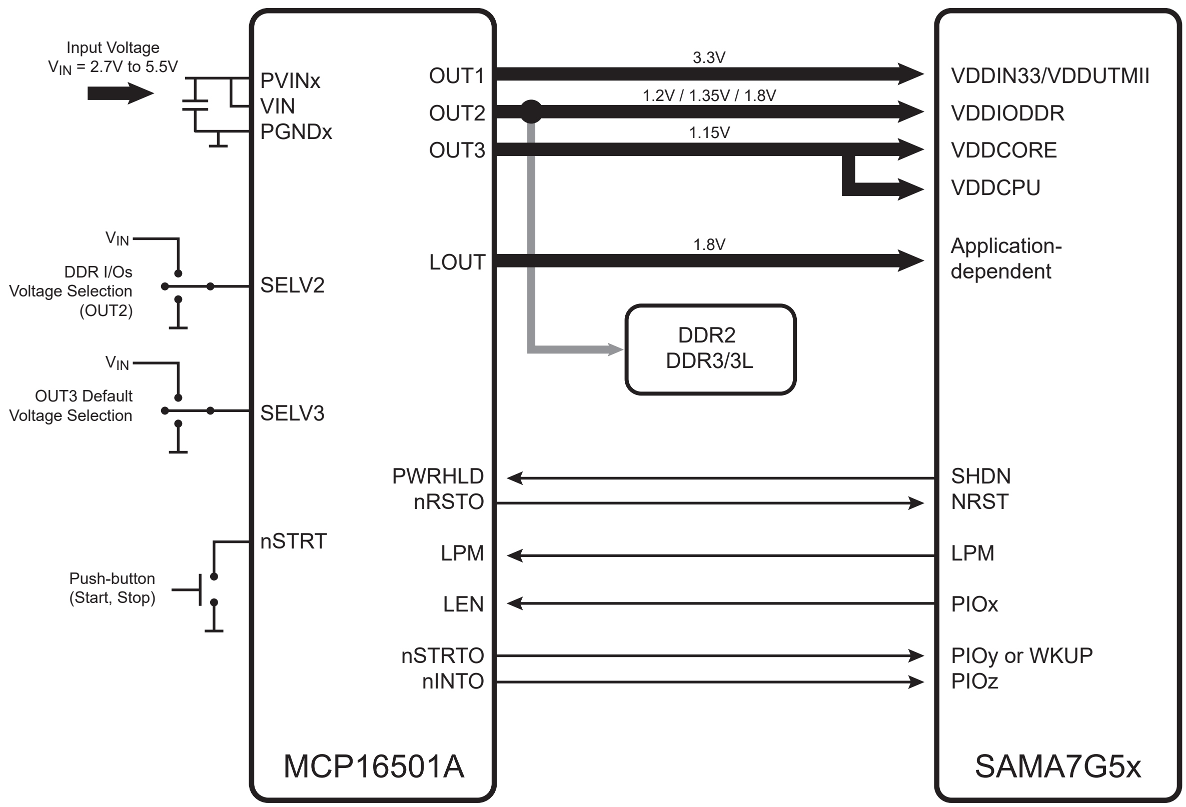
4.2 MCP16501 PMIC

MCP16501 is a 4-channel PMIC designed for PCB area constrained applications. It features three 1A DC-DC Buck regulators and one 0.3A auxiliary LDO regulator, and provides a simple, leakage-free interface with SAMA7G5. MCP16501 supports all SAMA7G5x low-power modes except ULP2, as VDDCPU is fed from the same power rail as VDDCORE. See the following figure.

- MCP16501**A** supports systems based on DDR2 (1.8V) or DDR3L (1.35V) SDRAM memories.
- MCP16501**D** supports applications using LPDDR2-SDRAM or LPDDR3-SDRAM. In this version, LOUT must be set at 1.8V and enabled by connecting LEN to OUT1.

For further details, refer to the MCP16501 documentation on www.microchip.com.

Figure 4-2. MCP16501 Simplified Application Block Diagram



5. Safety and Security Features

5.1 Design for Safety and IEC60730 Class B Certification

5.1.1 Background Information

The IEC 60730 standard encompasses all aspects of appliance design. Annex H of the standard covers the aspects most relevant to microcontrollers. It details the tests and diagnostics which are intended to ensure safe operation of embedded control hardware and software. IEC 60730 defines three classifications for electronic control functions:

- Class A - Control functions which are not intended to be relied upon for safety of the equipment
- Class B - Control functions intended to prevent unsafe operation of the controlled equipment
- Class C - Control functions intended to prevent special hazards such as explosions

Specific design techniques have been used in the SAMA7G5 to ease compliance with the IEC 60730 Class B Certification and to resolve general purpose safety concerns. This allows reduced software development and code size as well as savings on external hardware circuitry, since built-in self-tests are already embedded in the MPU. The table below gives the list of peripherals which incorporate these techniques, and details whether these features are applicable for the IEC 60730 Class B Certification or for general purpose safety considerations.

5.2 Design for Security

The SAMA7G5 embeds peripherals with security features to prevent counterfeiting, to secure external communication, and to authenticate the system.

The table below provides the list of peripherals and an overview of their security function. For more information, see the sections relevant to each peripheral.

5.3 Safety and IEC 60730 Features

Table 5-1. Safety and IEC 60730 Features List

Peripheral	Component	Fault/Error/Feature
PMC	Clock	CPU clock monitoring - Overclocking detection
		32.768 kHz crystal oscillator frequency monitoring - Abnormal frequency deviation
		Main crystal oscillator - Crystal failure detection
PIOC	I/O Periphery	Programmable configuration lock (active until next V_{DDCORE} reset) to protect against further software modifications (intentional or unintentional)
		Digital I/O - Plausibility check
ADCC		Analog I/O and ADC converter - Plausibility check
ICM (SHA)	Memory and Internal Data Path	All internal and external memories such as SMC, DDR, internal SRAM and QSPI
NAND Flash Controller ECC		Nonvolatile memory - Mutiple error detection (2 to 32)

...........continued

Peripheral	Component	Fault/Error/Feature
System Controller	Supply Monitor	Power supplies - VDDCPU, VDDCORE, VBAT abnormal levels
WDT, RSTC	Watchdog	Watchdog can be fed by an internal always on clock - Program counter stuck at faults.
		Watchdog configuration can be locked (write-protected until next reset) - Errant writes (Programming errors, errors introduced by system or hardware failures)
		Watchdog overflow generates a system reset
Cortex MMU	Memory Management Unit	Cortex-A7 Memory Management Unit
MATRIX, SYSC, ACC, PMC, PIO, SMC, SSC, I2SMCC, FLEXCOM, QSPI, TC, PDMC, ADC,ASRC, EIC, PWM, PIT64B, SPDIFRX, SPDIFTX, TZAESB, TZAESBASC, UHPHS	Peripherals	Configuration, Interrupt Enable/Disable, Control registers can be independently write-protected - Errant writes (Programming errors, errors introduced by system or hardware failures)
AES, TDES, SHA, PIT64B	Peripherals	Embedded integrity checker with reports in status registers.
AES, TDES	Peripherals	Immediate clear of keys in case of tamper detection. Immediate stop of processing in case of tamper detection
PWM, PIO	PWM	Fault inputs can be configured to put the PWM outputs in Safe mode - Programming errors, errors introduced by system or hardware failures
		PIO controller can lock the PWM I/O - Programming errors, errors introduced by system or hardware failures
		Fault inputs can be external (IO) or internal (ADC, TIMER, ACC, etc.) - Programming errors, errors introduced by system or hardware failures

5.4 Security Features

Table 5-2. Security Features

Peripheral	Function	Description	Comments
TrustZone	Security Enclave	Partition Secure/Non-secure world	Arm technology
Cortex MMU	Memory Management Unit	Cortex-A7 Memory Management Unit	–
PIO	I/O Control/ Peripheral Access	When a peripheral is not selected (PIO-controlled), IO lines have no access to the peripheral.	–
	Freeze	Capability to freeze either the functional part or the physical part of the configuration.	Once the freeze command is issued, no modifications to the current configuration are possible. Only a hardware reset allows a change to the configuration.

...........continued

Peripheral	Function	Description	Comments
Public Key Coprocessor (CPKCC) and associated Classical Public Key Cryptography Library (CPKCL)	Cryptography	ECC (Asymmetric key algorithm, elliptic curves)	–
		RSA (Asymmetric key algorithm)	
TDES, TRNG, AES, SHA		Hardware-accelerated Triple DES	FIPS-compliant[(1)]
		True Random Number Generator	
		Hardware-accelerated AES up to 256 bits	
		SHA up to 512 and HMAC-SHA	
AES, SHA, CPKCC, CPKLC	Secure Boot	Code encrypted/decrypted, Trusted Code Authentication	Hardware AES: Encrypt, Decrypt, CMAC Hardware SHA CPKCC, CPKCL: RSA or elliptic curves
AES, TDES, SHA, PIT64B, TC	Security and safety analysis and report	Monitoring on states or sequences, clocks and waveforms. Error detection can occur only in abnormal operating conditions.	–
	Register access protection	Checks for incorrect accesses.	–
AES, TDES	Key clearing on event	Immediate clearing of the key in case of external tamper event detection	–
TZAESB	On-the-fly AES	On-the-fly encryption/decryption for NFC_RAM, DDR, QSPI and SMC memories, with respect to TrustZone using TZAESBASC	AES128
TZAESBASC		Directs data transfer to either the TZAESB secure core or the unsecured TZAESB core	–
Private Key Bus	Transfers hidden keys to crypto-engines	Capability to transfer keys to or from AES, TZAESB, TDES, TRNG, OTPC in a manner totally invisible by the software.	–
Memories	Scrambling	On-the-fly scrambling/unscrambling for memories	SMC, SECURAM, GPBR and QSPI
ICM	Memory Integrity Check Monitoring	Uses a hardware Secure Hash Algorithm (up to SHA256)	SMC, DDR, internal SRAM and QSPI

...........continued

Peripheral	Function	Description	Comments
SECUMOD	JTAG	JTAG entry monitor	These tampers (JTAG, test, PIOBUs, monitors, etc.) can be configured to immediately erase Backup memories (BUSRAM4KB and BUREG256b) or generate an interrupt or a wakeup signal.
	Test	Test entry monitor	
	Voltage Monitoring	VBAT monitoring	
		VDDCPU monitoring	
		VDDCORE monitoring	
	Temperature Monitoring	Temperature monitoring	
	Frequency Monitoring	32.768 kHz crystal oscillator monitoring	
		CPU clock monitoring	
	IO Tamper Pin	4 tamper detection pins. Active and Dynamic modes supported.	
	Secure Backup SRAM (SECURAM)	5 Kbytes scrambled and non-imprinting avoiding data persistance	4 Kbytes erasable on tamper detection
	Secure Backup Registers	256-bit register bank, scrambled	Erasable on tamper detection
RTC	RTC	Timestamping of tamper events. Protection against bad configuration (invalid entry for date and time are impossible)	All events are logged in the RTC. Timestamping gives the source of the reset/ erase memory/interruption
		RTC robustness against glitch attack on 32 kHz crystal oscillator	–
Secure OTP	JTAG Access Control	Disable JTAG access by OTP bit	–
	Secure Debug Disable	JTAG debug allowed in Normal mode only, not in Secure mode	TrustZone
TZWDT	Watchdog	Protects against TrustZone starvation	TrustZone
GPBR	Peripheral Access and Protection	GPBR can be write protected, read protected and immediately cleared on external tamper event detection	–

Note:
1. Refer to each peripheral section for details on FIPS compliance.

6. Event System

A set of events generated by "source" peripherals are routed to "destination" peripherals without processor intervention.

6.1 Real-Time Event List

- Timers, PWM, and IO peripherals generate event triggers which are directly routed to event managers such as ADC, for example, to start measurement/conversion without processor intervention.
- ADC is connected to nine trigger inputs defined as two groups:
  – One group of eight elements for Timer Counter (TC0 to TC4), ADTRIG and PMW0 event0, PWM0 event1
  – One group of one element for low-rate trigger, RTC
- PWM safety events (faults) are in combinational form and directly routed from event generators (ADC, ACC, PMC, TIMER) to the PWM module.
- PWM output comparators generate events directly connected to TIMER.
- PMC safety event (clock failure detection) can be programmed to switch the MCK on a reliable main RC internal clock without processor intervention.

6.2 Real-Time Event Mapping

Table 6-1. Real-Time Event Mapping List

Function	Application	Description	Event Source	Event Destination
Safety	General-purpose	Automatic switch to reliable main RC oscillator in case of main crystal clock failure(1)	Power Management Controller (PMC)	PMC
	General-purpose, motor control, Power Factor Correction (PFC)	Puts the PWM outputs in Safe mode in case of main crystal clock failure(1)(2)	PMC	Pulse Width Modulation (PWM)
	Motor control, PFC	Puts the PWM outputs in Safe mode (overcurrent detection, etc.)(2)(3)	Analog-to-Digital Converter (ADC)	PWM
	Motor control	Puts the PWM outputs in Safe mode (overspeed detection through timer quadrature decoder)(2)(4)	TC0 PWM FAULT	PWM
	General-purpose, motor control, PFC	Puts the PWM outputs in Safe mode (general-purpose fault inputs)(2)	PWM_FIx	PWM

...........continued

Function	Application	Description	Event Source	Event Destination
Measurement trigger	Power factor correction (DC-DC, lighting, etc.)	Duty cycle output waveform correction Trigger source selection in PWM(5)(6)	ACC	PWM
			PWM_EXTRGx	PWM
	General-purpose	Trigger source selection in ADCC(7)	PIO ADTRG	ADCC
			TC0 TIOA0	ADCC
			TC0 TIOA1	ADCC
			TC0 TIOA2	ADCC
			TC1 TIOA3	ADCC
			ACC	ADCC
	Motor control	ADC-PWM synchronization(10)(11) Trigger source selection in ADCC(7)	PWM event lines 0 and 1	ADCC
	General-purpose	Temperature sensor Low-speed measurement(10)(11)	RTC RTCOUT0	ADCC
Delay measurement	Motor control	Propagation delay of external components (IOs, power transistor bridge driver, etc.)(12)(13)	PWM comparator output OC0	TC0, TIOA0 and TIOB0
			PWM comparator output OC1	TC0, TIOA1 and TIOB1
			PWM comparator output OC2	TC0, TIOA2 and TIOB2
Audio clock recovery from Ethernet	Audio	GMAC GTSUCOMP signal adaptation via TC (TC_EMR.TRIGSRCB) in order to drive the clock reference of the external PLL for the audio clock	GMAC0 GTSUCOMP	TC1 TIOB1
Direct Memory Access	General-purpose	Peripheral trigger event generation to transfer data to/from system memory(14)	Peripherals with DMA HW interface number	XDMA0
				XDMA1
Wake-up	Wake On LAN	GMAC uses WOL frame comparison and wake-up the core by enabling the crystal	GMAC0, GMAC1	PMC

Notes:

1. Refer to Main Crystal Oscillator Failure Detection in the section Power Management Controller (PMC).
2. Refer to Fault Inputs and Fault Protection in the section Pulse Width Modulation Controller (PWM).
3. Refer to Fault Mode in the section Analog Comparator Controller (ACC).
4. Refer to Fault Mode in the section Analog Comparator Controller (ACC).
5. Refer to PWM_ETRGx in the section Pulse Width Modulation Controller (PWM).
6. Refer to PWM External Trigger Mode in the section Pulse Width Modulation Controller (PWM).
7. Refer to Conversion Triggers in the section Analog-to-Digital Converter (ADC).
8. Refer to Temperature Sensor in the section Analog-to-Digital Converter (ADC).
9. Refer to Waveform Generation in the section Real-time Clock (RTC).
10. Refer to PWM_CMPVx in the section Pulse Width Modulation Controller (PWM).
11. Refer to PWM Comparison Units and PWM Event Lines in the section Pulse Width Modulation Controller (PWM).
12. Refer to Comparator in the section Pulse Width Modulation Controller (PWM).
13. Refer to Synchronization with PWM in the section Timer Counter (TC).
14. Refer to the section DMA Controller (XDMAC).

7. Package and Pinout

7.1 Package

The SAMA7G5 is available in the following package:

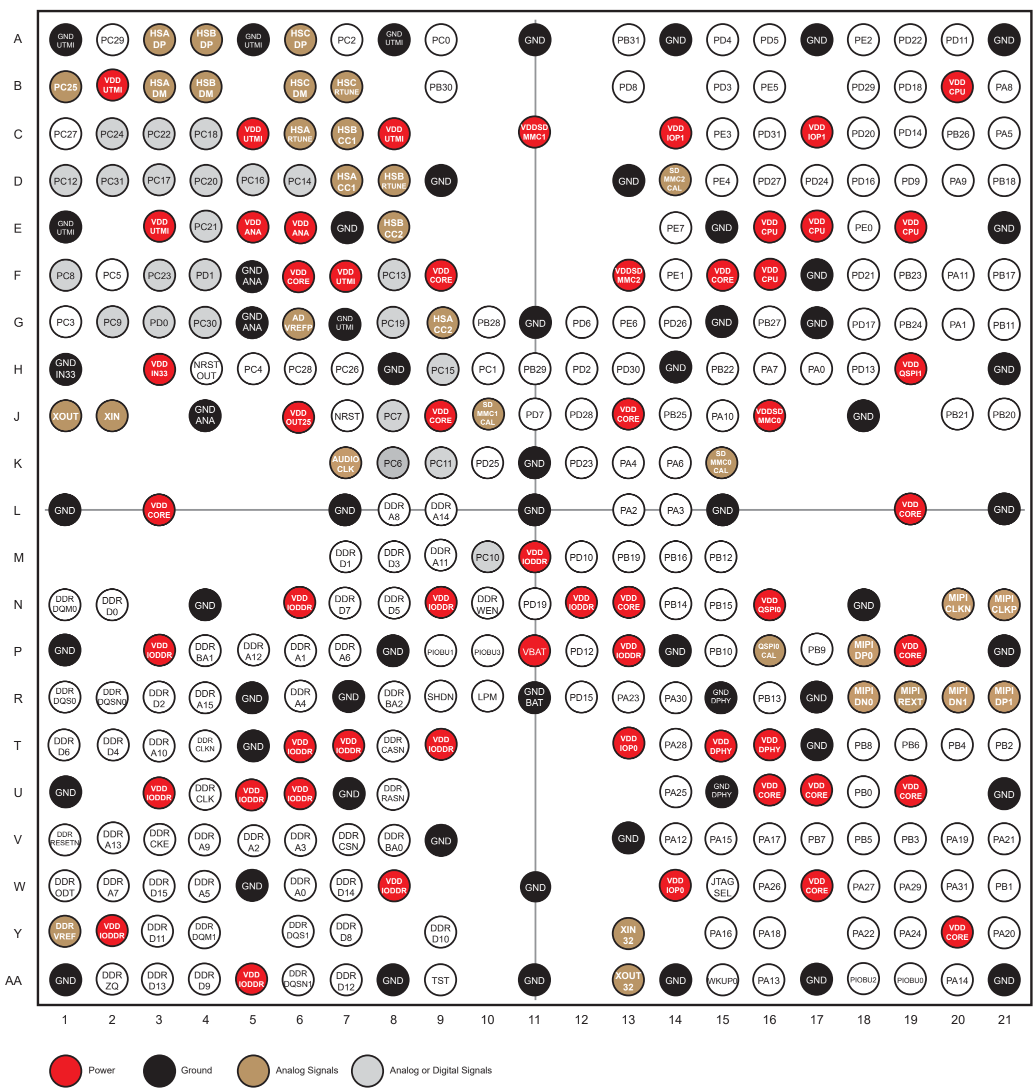
- 14x14 mm^2, 0.65-mm pitch 343-ball TFBGA optimized for 4-layer PCB

For further details, refer to Mechanical Characteristics.

7.2 Pinout

7.2.1 BGA343 Pinout

Figure 7-1. 343-ball TFBGA Pinout



The device features several PIO controllers that multiplex the I/O lines of the peripheral set. The following Pin Description table defines how the I/O lines are multiplexed on the different PIO controllers. The "Reset State" column shows whether the PIO line resets in I/O mode or in Peripheral mode. If I/O is shown, the PIO line resets with the characteristics (input, output, pull-up or pull-down) indicated in this same column, so that the device is configured in a known state as soon as the reset is released. As a result, PIO_CFGR.FUNC resets to ‘0’. If a signal name is shown in the “Reset State” column, the PIO line is assigned to this function and PIO_CFGR.FUNC is not set to ‘0’. That is the case for pins controlling memories, in particular address lines, which require the pin to be driven as soon as the reset is released.

Table 7-1. Pin Description[(1)(2)]

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
H17	VDDSDMMC0	HSIO	PA0	I/O	–	–	A	SDMMC0_CK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO0	I/O	1	
							C	CANTX3	O	1	
							E	PWML0	O	3	
G20	VDDSDMMC0	HSIO	PA1	I/O	–	–	A	SDMMC0_CMD	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO1	I/O	1	
							C	CANRX3	I	1	
							D	D14	I/O	1,2	
							E	PWMH0	O	3	
L13	VDDSDMMC0	GPIO	PA2	I/O	–	–	A	SDMMC0_RSTN	O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO2	I/O	1	
							C	PDMC1_CLK	O	1	
							D	D15	I/O	1,2	
							E	PWMH1	O	3	
							F	FLEXCOM1_IO0	I/O	3	
L14	VDDSDMMC0	HSIO	PA3	I/O	–	–	A	SDMMC0_DAT0	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO3	I/O	1	
							C	PDMC1_DS0	I	1	
							D	NWR1/NBS1	O	1,2	
							E	PWML3	O	3	
							F	FLEXCOM1_IO1	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
K13	VDDSDMMC0	HSIO	PA4	I/O	–	–	A	SDMMC0_DAT1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO4	I/O	1	
							C	PDMC1_DS1	I	1	
							D	NCS2	O	1,2	
							E	PWMH3	O	3	
							F	FLEXCOM2_IO0	I/O	3	
C21	VDDSDMMC0	HSIO	PA5	I/O	–	–	A	SDMMC0_DAT2	I/O	1	PIO, I, PU, ST
							B	FLEXCOM1_IO0	I/O	1	
							C	CANTX2	O	1	
							D	A23	O	1,2	
							E	PWMEXTRG0	I	3	
							F	FLEXCOM2_IO1	I/O	3	
K14	VDDSDMMC0	HSIO	PA6	I/O	–	–	A	SDMMC0_DAT3	I/O	1	PIO, I, PU, ST
							B	FLEXCOM1_IO1	I/O	1	
							C	CANRX2	I	1	
							D	A24	O	1,2	
							E	PWMEXTRG1	I	3	
							F	FLEXCOM3_IO0	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
H16	VDDSDMMC0	HSIO	PA7	I/O	–	–	A	SDMMC0_DAT4	I/O	1	PIO, I, PU, ST
							B	FLEXCOM2_IO0	I/O	1	
							C	CANTX1	O	1	
							D	NWAIT	I	1,2	
							E	PWMFI0	I	3	
							F	FLEXCOM3_IO1	I/O	3	
B21	VDDSDMMC0	HSIO	PA8	I/O	–	–	A	SDMMC0_DAT5	I/O	1	PIO, I, PU, ST
							B	FLEXCOM2_IO1	I/O	1	
							C	CANRX1	I	1	
							D	NCS0	O	1,2	
							E	PWMFI1	I	3	
							F	FLEXCOM4_IO0	I/O	3	
D20	VDDSDMMC0	HSIO	PA9	I/O	–	–	A	SDMMC0_DAT6	I/O	1	PIO, I, PU, ST
							B	FLEXCOM2_IO2	I/O	1	
							C	CANTX0	O	1	
							D	SMCK	O	1,2	
							E	SPDIF_RX	I	1	
							F	FLEXCOM4_IO1	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
J15	VDDSDMMC0	HSIO	PA10	I/O	–	–	A	SDMMC0_DAT7	I/O	1	PIO, I, PU, ST
							B	FLEXCOM2_IO3	I/O	1	
							C	CANRX0	I	1	
							D	NCS1	O	1,2	
							E	SPDIF_TX	O	1	
							F	FLEXCOM5_IO0	I/O	3	
F20	VDDSDMMC0	HSIO	PA11	I/O	–	–	A	SDMMC0_DS	I	1	PIO, I, PU, ST
							B	FLEXCOM2_IO4	I/O	1	
							D	A0/NBS0	O	1,2	
							E	TIOA0	I/O	1	
							F	FLEXCOM5_IO1	I/O	3	
V14	VDDIOP0	GPIO	PA12	I/O	–	–	A	SDMMC0_WP	I	1	PIO, I, PU, ST
							B	FLEXCOM1_IO3	I/O	1	
							D	FLEXCOM3_IO5	I/O	1	
							E	PWML2	O	3	
							F	FLEXCOM6_IO0	I/O	3	
AA16	VDDIOP0	GPIO	PA13	I/O	–	–	A	SDMMC0_1V8SEL	O	1	PIO, I, PU, ST
							B	FLEXCOM1_IO2	I/O	1	
							D	FLEXCOM3_IO6	I/O	1	
							E	PWMH2	O	3	
							F	FLEXCOM6_IO1	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
AA20	VDDIOP0	GPIO	PA14	I/O	–	–	A	SDMMC0_CD	I	1	PIO, I, PU, ST
							B	FLEXCOM1_IO4	I/O	1	
							D	A25	O	1,2	
							E	PWML1	O	3	
V15	VDDIOP0	GPIO	PA15	I/O	–	–	A	G0_TXEN	O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO0	I/O	1	
							C	ISC_MCK	O	1	
							D	A1	O	1,2	
							E	TIOB0	I/O	1	
Y15	VDDIOP0	GPIO	PA16	I/O	–	–	A	G0_TX0	O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO1	I/O	1	
							C	ISC_D0	I	1	
							D	A2	O	1,2	
							E	TCLK0	I	1	
V16	VDDIOP0	GPIO	PA17	I/O	–	–	A	G0_TX1	O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO2	I/O	1	
							C	ISC_D1	I	1	
							D	A3	O	1,2	
							E	TIOA1	I/O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
Y16	VDDIOP0	GPIO	PA18	I/O	–	–	A	G0_RXDV	I	1	PIO, I, PU, ST
							B	FLEXCOM3_IO3	I/O	1	
							C	ISC_D2	I	1	
							D	A4	O	1,2	
							E	TIOB1	I/O	1	
V20	VDDIOP0	GPIO	PA19	I/O	–	–	A	G0_RX0	I	1	PIO, I, PU, ST
							B	FLEXCOM3_IO4	I/O	1	
							C	ISC_D3	I	1	
							D	A5	O	1,2	
							E	TCLK1	I	1	
Y21	VDDIOP0	GPIO	PA20	I/O	–	–	A	G0_RX1	I	1	PIO, I, PU, ST
							B	FLEXCOM4_IO0	I/O	1	
							C	ISC_D4	I	1	
							D	A6	O	1,2	
							E	TIOA2	I/O	1	
V21	VDDIOP0	GPIO	PA21	I/O	–	–	A	G0_RXER	I	1	PIO, I, PU, ST
							B	FLEXCOM4_IO1	I/O	1	
							C	ISC_D5	I	1	
							D	A7	O	1,2	
							E	TIOB2	I/O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
Y18	VDDIOP0	GPIO	PA22	I/O	–	–	A	G0_MDC	O	1	PIO, I, PU, ST
							B	FLEXCOM4_IO2	I/O	1	
							C	ISC_D6	I	1	
							D	A8	O	1,2	
							E	TCLK2	I	1	
R13	VDDIOP0	GPIO	PA23	I/O	–	–	A	G0_MDIO	I/O	1	PIO, I, PU, ST
							B	FLEXCOM4_IO3	I/O	1	
							C	ISC_D7	I	1	
							D	A9	O	1,2	
Y19	VDDIOP0	GPIO	PA24	I/O	–	–	A	G0_TXCK/ G0_REFCK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM4_IO4	I/O	1	
							C	ISC_HSYNC	I	1	
							D	A10	O	1,2	
							E	FLEXCOM0_IO5	I/O	1	
U14	VDDIOP0	GPIO	PA25	I/O	–	–	A	G0_125CK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM5_IO4	I/O	1	
							C	ISC_VSYNC	I	1	
							D	A11	O	1,2	
							E	FLEXCOM0_IO6	I/O	1	
							F	FLEXCOM7_IO0	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
W16	VDDIOP0	GPIO	PA26	I/O	–	–	A	G0_TX2	O	1	PIO, I, PU, ST
							B	FLEXCOM5_IO2	I/O	1	
							C	ISC_FIELD	I	1	
							D	A12	O	1,2	
							E	TF0	I/O	1	
							F	FLEXCOM7_IO1	I/O	3	
W18	VDDIOP0	GPIO	PA27	I/O	–	–	A	G0_TX3	O	1	PIO, I, PU, ST
							B	FLEXCOM5_IO3	I/O	1	
							C	ISC_PCK	I	1	
							D	A13	O	1,2	
							E	TK0	I/O	1	
							F	FLEXCOM8_IO0	I/O	3	
T14	VDDIOP0	GPIO	PA28	I/O	–	–	A	G0_RX2	I	1	PIO, I, PU, ST
							B	FLEXCOM5_IO0	I/O	1	
							C	ISC_D8	I	1	
							D	A14	O	1,2	
							E	RD0	I	1	
							F	FLEXCOM8_IO1	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
W19	VDDIOP0	GPIO	PA29	I/O	–	–	A	G0_RX3	I	1	PIO, I, PU, ST
							B	FLEXCOM5_IO1	I/O	1	
							C	ISC_D9	I	1	
							D	A15	O	1,2	
							E	RF0	I/O	1	
							F	FLEXCOM9_IO0	I/O	3	
R14	VDDIOP0	GPIO	PA30	I/O	–	–	A	G0_RXCK	I	1	PIO, I, PU, ST
							B	FLEXCOM6_IO4	I/O	1	
							C	ISC_D10	I	1	
							D	A16	O	1,2	
							E	RK0	I/O	1	
							F	FLEXCOM9_IO1	I/O	3	
W20	VDDIOP0	GPIO	PA31	I/O	–	–	A	G0_TXER	O	1	PIO, I, PU, ST
							B	FLEXCOM6_IO2	I/O	1	
							C	ISC_D11	I	1	
							D	A17	O	1,2	
							E	TD0	O	1	
							F	FLEXCOM10_IO0	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
U18	VDDIOP0	GPIO	PB0	I/O	–	–	A	G0_COL	I	1	PIO, I, PU, ST
							B	FLEXCOM6_IO3	I/O	1	
							C	EXT_IRQ0	I	1	
							D	A18	O	1,2	
							E	SPDIF_RX	I	2	
							F	FLEXCOM10_IO1	I/O	3	
W21	VDDIOP0	GPIO	PB1	I/O	–	–	A	G0_CRS	I	1	PIO, I, PU, ST
							B	FLEXCOM6_IO1	I/O	1	
							C	EXT_IRQ1	I	1	
							D	A19	O	1,2	
							E	SPDIF_TX	O	2	
							F	FLEXCOM11_IO0	I/O	3	
T21	VDDIOP0	GPIO	PB2	I/O	–	–	A	G0_TSUCOMP	O	1	PIO, I, PU, ST
							B	FLEXCOM6_IO0	I/O	1	
							C	ADTRG	I	1	
							D	A20	O	1,2	
							F	FLEXCOM11_IO1	I/O	3	
V19	VDDIOP0	GPIO	PB3	I/O	–	–	A	RF1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM11_IO0	I/O	1	
							C	PCK2	O	2	
							D	D8	I/O	1,2	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
T20	VDDIOP0	GPIO	PB4	I/O	–	–	A	TF1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM11_IO1	I/O	1	
							C	PCK3	O	2	
							D	D9	I/O	1,2	
V18	VDDIOP0	GPIO	PB5	I/O	–	–	A	TK1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM11_IO2	I/O	1,2,3,4,5	
							C	PCK4	O	2	
							D	D10	I/O	1,2	
T19	VDDIOP0	GPIO	PB6	I/O	–	–	A	RK1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM11_IO3	I/O	1,2,3,4,5	
							C	PCK5	O	2	
							D	D11	I/O	1,2	
V17	VDDIOP0	GPIO	PB7	I/O	–	–	A	TD1	O	1	PIO, I, PU, ST
							B	FLEXCOM11_IO4	I/O	1,2,3,4,5	
							C	FLEXCOM3_IO5	I/O	2,3,4,5	
							D	D12	I/O	1,2	
T18	VDDIOP0	GPIO	PB8	I/O	–	–	A	RD1	I	1	PIO, I, PU, ST
							B	FLEXCOM8_IO0	I/O	1	
							C	FLEXCOM3_IO6	I/O	2,3,4,5	
							D	D13	I/O	1,2	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
P17	VDDQSPI0	HSIO	PB9	I/O	–	–	A	QSPI0_IO3	I/O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO1	I/O	1	
							C	PDMC0_CLK	O	1	
							D	NCS3/NANDCS	O	1	
							E	PWML0	O	2	
P15	VDDQSPI0	HSIO	PB10	I/O	–	–	A	QSPI0_IO2	I/O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO2	I/O	1	
							C	PDMC0_DS0	I	1	
							D	NWE/NWR0/ NANDWE	O	1	
							E	PWMH0	O	2	
G21	VDDQSPI0	HSIO	PB11	I/O	–	–	A	QSPI0_IO1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO3	I/O	1	
							C	PDMC0_DS1	I	1	
							D	NRD/NANDOE	O	1	
							E	PWML1	O	2	
M15	VDDQSPI0	HSIO	PB12	I/O	–	–	A	QSPI0_IO0	I/O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO4	I/O	1	
							C	FLEXCOM6_IO5	I/O	1	
							D	A21/NANDALE	O	1	
							E	PWMH1	O	2	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
R16	VDDQSPI0	GPIO	PB13	I/O	–	–	A	QSPI0_CS	O	1	PIO, I, PU, ST
							B	FLEXCOM9_IO0	I/O	1	
							C	FLEXCOM6_IO6	I/O	1	
							D	A22/NANDCLE	O	1	
							E	PWML2	O	2	
N14	VDDQSPI0	HSIO	PB14	I/O	–	–	A	QSPI0_SCK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM9_IO1	I/O	1	
							D	D0	I/O	1	
							E	PWMH2	O	2	
N15	VDDQSPI0	HSIO	PB15	I/O	–	–	A	QSPI0_SCKN	I/O	1	PIO, I, PU, ST
							B	FLEXCOM9_IO2	I/O	1	
							D	D1	I/O	1	
							E	PWML3	O	2	
M14	VDDQSPI0	HSIO	PB16	I/O	–	–	A	QSPI0_IO4	I/O	1	PIO, I, PU, ST
							B	FLEXCOM9_IO3	I/O	1	
							C	PCK0	O	1	
							D	D2	I/O	1	
							E	PWMH3	O	2	
							F	EXT_IRQ0	I	2	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
F21	VDDQSPI0	HSIO	PB17	I/O	–	–	A	QSPI0_IO5	I/O	1	PIO, I, PU, ST
							B	FLEXCOM9_IO4	I/O	1	
							C	PCK1	O	1	
							D	D3	I/O	1	
							E	PWMEXTRG0	I	2	
							F	EXT_IRQ1	I	2	
D21	VDDQSPI0	HSIO	PB18	I/O	–	–	A	QSPI0_IO6	I/O	1	PIO, I, PU, ST
							B	FLEXCOM10_IO0	I/O	1	
							C	PCK2	O	1	
							D	D4	I/O	1	
							E	PWMEXTRG1	I	2	
M13	VDDQSPI0	HSIO	PB19	I/O	–	–	A	QSPI0_IO7	I/O	1	PIO, I, PU, ST
							B	FLEXCOM10_IO1	I/O	1	
							C	PCK3	O	1	
							D	D5	I/O	1	
							E	PWMFI0	I	2	
J21	VDDQSPI0	HSIO	PB20	I/O	–	–	A	QSPI0_DQS	I	1	PIO, I, PU, ST
							B	FLEXCOM10_IO2	I/O	1,2,3,4,5	
							D	D6	I/O	1	
							E	PWMFI1	I	2	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
J20	VDDQSPI0	GPIO	PB21	I/O	–	–	A	QSPI0_INT	I	1	PIO, I, PU, ST
							B	FLEXCOM10_IO3	I/O	1,2,3,4,5	
							C	FLEXCOM9_IO5	I/O	1	
							D	D7	I/O	1	
H15	VDDQSPI1	GPIO	PB22	I/O	–	–	A	QSPI1_IO3	I/O	1	PIO, I, PU, ST
							B	FLEXCOM10_IO4	I/O	1,2,3,4,5	
							C	FLEXCOM9_IO6	I/O	1	
							D	NANDRDY	I	1	
F19	VDDQSPI1	GPIO	PB23	I/O	–	–	A	QSPI1_IO2	I/O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO0	I/O	1	
							C	I2SMCC0_CK	I/O	1	
							F	PCK4	O	1	
G19	VDDQSPI1	GPIO	PB24	I/O	–	–	A	QSPI1_IO1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO1	I/O	1	
							C	I2SMCC0_WS	I/O	1	
							F	PCK5	O	1	
J14	VDDQSPI1	GPIO	PB25	I/O	–	–	A	QSPI1_IO0	I/O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO2	I/O	1	
							C	I2SMCC0_DOUT1	O	1	
							F	PCK6	O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
C20	VDDQSPI1	GPIO	PB26	I/O	–	–	A	QSPI1_CS	O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO3	I/O	1	
							C	I2SMCC0_DOUT0	O	1	
							E	PWMEXTRG0	I	1	
							F	PCK7	O	1	
G16	VDDQSPI1	GPIO	PB27	I/O	–	–	A	QSPI1_SCK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO4	I/O	1	
							C	I2SMCC0_MCK	O	1	
							E	PWMEXTRG1	I	1	
G10	VDDSDMMC1	GPIO	PB28	I/O	–	–	A	SDMMC1_RSTN	O	1	PIO, I, PU, ST
							B	ADTRG	I	2	
							E	PWMFI0	I	1	
							F	FLEXCOM7_IO0	I/O	4	
H11	VDDSDMMC1	HSIO	PB29	I/O	–	–	A	SDMMC1_CMD	I/O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO2	I/O	2,3,4,5	
							C	FLEXCOM0_IO5	I/O	2	
							D	TIOA3	I/O	1	
							E	PWMFI1	I	1	
							F	FLEXCOM7_IO1	I/O	4	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B9	VDDSDMMC1	HSIO	PB30	I/O	–	–	A	SDMMC1_CK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO3	I/O	2,3,4,5	
							C	FLEXCOM0_IO6	I/O	2	
							D	TIOB3	I/O	1	
							E	PWMH0	O	1	
							F	FLEXCOM8_IO0	I/O	4	
A13	VDDSDMMC1	HSIO	PB31	I/O	–	–	A	SDMMC1_DAT0	I/O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO4	I/O	2,3,4,5	
							C	FLEXCOM9_IO5	I/O	2,3,4,5	
							D	TCLK3	I	1	
							E	PWML0	O	1	
							F	FLEXCOM8_IO1	I/O	4	
A9	VDDSDMMC1	HSIO	PC0	I/O	–	–	A	SDMMC1_DAT1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO0	I/O	2	
							D	TIOA4	I/O	1	
							E	PWML1	O	1	
							F	FLEXCOM9_IO0	I/O	4	
H10	VDDSDMMC1	HSIO	PC1	I/O	–	–	A	SDMMC1_DAT2	I/O	1	PIO, I, PU, ST
							B	FLEXCOM3_IO1	I/O	2	
							D	TIOB4	I/O	1	
							E	PWMH1	O	1	
							F	FLEXCOM9_IO1	I/O	4	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A7	VDDSDMMC1	HSIO	PC2	I/O	–	–	A	SDMMC1_DAT3	I/O	1	PIO, I, PU, ST
							B	FLEXCOM4_IO0	I/O	2	
							D	TCLK4	I	1	
							E	PWML2	O	1	
							F	FLEXCOM10_IO0	I/O	4	
G1	VDDIN33	GPIO	PC3	I/O	–	–	A	SDMMC1_WP	I	1	PIO, I, PU, ST
							B	FLEXCOM4_IO1	I/O	2	
							D	TIOA5	I/O	1	
							E	PWMH2	O	1	
							F	FLEXCOM10_IO1	I/O	4	
H5	VDDIN33	GPIO	PC4	I/O	–	–	A	SDMMC1_CD	I	1	PIO, I, PU, ST
							B	FLEXCOM4_IO2	I/O	2,3,4,5	
							C	FLEXCOM9_IO6	I/O	2,3,4,5	
							D	TIOB5	I/O	1	
							E	PWML3	O	1	
							F	FLEXCOM11_IO0	I/O	4	
F2	VDDIN33	GPIO	PC5	I/O	–	–	A	SDMMC1_1V8SEL	O	1	PIO, I, PU, ST
							B	FLEXCOM4_IO3	I/O	2,3,4,5	
							C	FLEXCOM6_IO5	I/O	2,3,4,5	
							D	TCLK5	I	1	
							E	PWMH3	O	1	
							F	FLEXCOM11_IO1	I/O	4	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
K8	VDDIN33	GPIO	PC6	I/O	ACC_INP0	I	A	–			PIO, I, PU, ST
							B	FLEXCOM4_IO4	I/O	2,3,4,5	
							C	FLEXCOM6_IO6	I/O	2,3,4,5	
J8	VDDIN33	GPIO	PC7	I/O	ACC_INN1	I	A	I2SMCC0_DIN0	I	1	PIO, I, PU, ST
							B	FLEXCOM7_IO0	I/O	2	
F1	VDDIN33	GPIO	PC8	I/O	ACC_INP1	I	A	I2SMCC0_DIN1	I	1	PIO, I, PU, ST
							B	FLEXCOM7_IO1	I/O	2	
G2	VDDIN33	GPIO	PC9	I/O	ACC_INN2	I	A	I2SMCC0_DOUT3	O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO2	I/O	2,3,4,5	
							F	FLEXCOM1_IO0	I/O	4	
M10	VDDIN33	GPIO	PC10	I/O	ACC_INP2	I	A	I2SMCC0_DOUT2	O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO3	I/O	2,3,4,5	
							F	FLEXCOM1_IO1	I/O	4	
K9	VDDIN33	GPIO	PC11	I/O	ACC_INN3	I	A	I2SMCC1_CK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM7_IO4	I/O	2,3,4,5	
							F	FLEXCOM2_IO0	I/O	4	
D1	VDDIN33	GPIO	PC12	I/O	ACC_INP3	I	A	I2SMCC1_WS	I/O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO2	I/O	2,3,4,5	
							F	FLEXCOM2_IO1	I/O	4	
F8	VDDIN33	GPIO	PC13	I/O	AD0	I	A	I2SMCC1_MCK	O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO1	I/O	2	
							F	FLEXCOM3_IO0	I/O	4	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
D6	VDDIN33	GPIO	PC14	I/O	AD1	I	A	I2SMCC1_DOUT0	O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO0	I/O	2	
							F	FLEXCOM3_IO1	I/O	4	
H9	VDDIN33	GPIO	PC15	I/O	AD2	I	A	I2SMCC1_DOUT1	O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO3	I/O	2,3,4,5	
							F	FLEXCOM4_IO0	I/O	4	
D5	VDDIN33	GPIO	PC16	I/O	AD3	I	A	I2SMCC1_DOUT2	O	1	PIO, I, PU, ST
							B	FLEXCOM8_IO4	I/O	2,3,4,5	
							F	FLEXCOM4_IO1	I/O	4	
D3	VDDIN33	GPIO	PC17	I/O	AD4	I/O	A	I2SMCC1_DOUT3	O	1	PIO, I, PU, ST
							B	EXT_IRQ0	I	3	
							F	FLEXCOM5_IO0	I/O	4	
C4	VDDIN33	GPIO	PC18	I/O	AD5	I/O	A	I2SMCC1_DIN0	I	1	PIO, I, PU, ST
							B	FLEXCOM9_IO0	I/O	2	
							F	FLEXCOM5_IO1	I/O	4	
G8	VDDIN33	GPIO	PC19	I/O	AD6	I	A	I2SMCC1_DIN1	I	1	PIO, I, PU, ST
							B	FLEXCOM9_IO1	I/O	2	
							F	FLEXCOM6_IO0	I/O	4	
D4	VDDIN33	GPIO	PC20	I/O	AD7	I	A	I2SMCC1_DIN2	I	1	PIO, I, PU, ST
							B	FLEXCOM9_IO4	I/O	2,3,4,5	
							F	FLEXCOM6_IO1	I/O	4	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
E4	VDDIN33	GPIO	PC21	I/O	AD8	I	A	I2SMCC1_DIN3	I	1	PIO, I, PU, ST
							B	FLEXCOM9_IO2	I/O	2,3,4,5	
							D	D3	I/O	2	
							F	FLEXCOM6_IO0	I/O	5	
C3	VDDIN33	GPIO	PC22	I/O	AD9	I	A	I2SMCC0_DIN2	I	1	PIO, I, PU, ST
							B	FLEXCOM9_IO3	I/O	2,3,4,5	
							D	D4	I/O	2	
							F	FLEXCOM6_IO1	I/O	5	
F3	VDDIN33	GPIO	PC23	I/O	AD10	I	A	I2SMCC0_DIN3	I	1	PIO, I, PU, ST
							B	FLEXCOM0_IO5	I/O	3	
							D	D5	I/O	2	
							F	FLEXCOM7_IO0	I/O	5	
C2	VDDIN33	GPIO	PC24	I/O	AD11	I	A	–			PIO, I, PU, ST
							B	FLEXCOM0_IO6	I/O	3	
							C	EXT_IRQ1	I	3	
							D	D6	I/O	2	
							F	FLEXCOM7_IO1	I/O	5	
B1	VDDIN33	GPIO	PC25	I/O	–	–	A	NTRST	I	1	NTRST, PU, ST
H7	VDDIN33	GPIO	PC26	I/O	–	–	A	TCK_SWCLK	I	1	TCK_SWCLK, ST
C1	VDDIN33	GPIO	PC27	I/O	–	–	A	TMS_SWDIO	I/O	1	TMS_SWDIO, PU, ST
H6	VDDIN33	GPIO	PC28	I/O	–	–	A	TDI	I	1	TDI, PU, ST

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A2	VDDIN33	GPIO	PC29	I/O	–	–	A	TDO	O	1	TDO, ST
G4	VDDIN33	GPIO	PC30	I/O	AD12	I	A	–			PIO, I, PD, ST
							B	FLEXCOM10_IO0	I/O	2	
D2	VDDIN33	GPIO	PC31	I/O	AD13	I	A	–			PIO, I, PD, ST
							B	FLEXCOM10_IO1	I/O	2	
G3	VDDIN33	GPIO	PD0	I/O	AD14	I	A	–			PIO, I, PD, ST
							B	FLEXCOM11_IO0	I/O	2	
F4	VDDIN33	GPIO	PD1	I/O	AD15	I	A	–			PIO, I, PD, ST
							B	FLEXCOM11_IO1	I/O	2	
H12	VDDSDMMC2	GPIO	PD2	I/O	–	–	A	SDMMC2_RSTN	O	1	PIO, I, PU, ST
							B	PCK0	O	2	
							C	CANTX4	O	1	
							D	D7	I/O	2	
							E	TIOA0	I/O	2	
							F	FLEXCOM8_IO0	I/O	5	
B15	VDDSDMMC2	HSIO	PD3	I/O	–	–	A	SDMMC2_CMD	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO0	I/O	2	
							C	CANRX4	I	1	
							D	NANDRDY	I	2	
							E	TIOB0	I/O	2	
							F	FLEXCOM8_IO1	I/O	5	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A15	VDDSDMMC2	HSIO	PD4	I/O	–	–	A	SDMMC2_CK	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO1	I/O	2	
							C	CANTX5	O	1	
							D	NCS3/NANDCS	O	2	
							E	TCLK0	I	2	
							F	FLEXCOM9_IO0	I/O	5	
A16	VDDSDMMC2	HSIO	PD5	I/O	–	–	A	SDMMC2_DAT0	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO2	I/O	2,3	
							C	CANRX5	I	1	
							D	NWE/NWR0/ NANDWE	O	2	
							E	TIOA1	I/O	2	
							F	FLEXCOM9_IO1	I/O	5	
G12	VDDSDMMC2	HSIO	PD6	I/O	–	–	A	SDMMC2_DAT1	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO3	I/O	2,3	
							C	SPDIF_RX	I	3	
							D	NRD/NANDOE	O	2	
							E	TIOB1	I/O	2	
							F	FLEXCOM10_IO0	I/O	5	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
J11	VDDSDMMC2	HSIO	PD7	I/O	–	–	A	SDMMC2_DAT2	I/O	1	PIO, I, PU, ST
							B	FLEXCOM0_IO4	I/O	2,3	
							C	SPDIF_TX	O	3	
							D	A21/NANDALE	O	2	
							E	TCLK1	I	2	
							F	FLEXCOM10_IO1	I/O	5	
B13	VDDSDMMC2	HSIO	PD8	I/O	–	–	A	SDMMC2_DAT3	I/O	1	PIO, I, PU, ST
							C	I2SMCC0_DIN0	I	2	
							D	A22/NANDCLE	O	2	
							E	TIOA2	I/O	2	
							F	FLEXCOM11_IO0	I/O	5	
D19	VDDIOP1	GPIO	PD9	I/O	–	–	A	SDMMC2_WP	I	1	PIO, I, PU, ST
							C	I2SMCC0_DIN1	I	2	
							D	D0	I/O	2	
							E	TIOB2	I/O	2	
							F	FLEXCOM11_IO1	I/O	5	
M12	VDDIOP1	GPIO	PD10	I/O	–	–	A	SDMMC2_CD	I	1	PIO, I, PU, ST
							B	PCK6	O	2	
							C	I2SMCC0_DIN2	I	2	
							D	D1	I/O	2	
							E	TCLK2	I	2	
							F	FLEXCOM0_IO0	I/O	3	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A20	VDDIOP1	GPIO	PD11	I/O	–	–	A	SDMMC2_1V8SEL	O	1	PIO, I, PU, ST
							B	PCK7	O	2	
							C	I2SMCC0_DIN3	I	2	
							D	D2	I/O	2	
							E	TIOA3	I/O	2	
							F	FLEXCOM0_IO1	I/O	3	
P12	VDDIOP1	GPIO	PD12	I/O	–	–	A	PCK1	O	2	PIO, I, PU, ST
							B	FLEXCOM1_IO0	I/O	2	
							D	CANTX0	O	2	
							E	TIOB3	I/O	2	
H18	VDDIOP1	GPIO	PD13	I/O	–	–	A	I2SMCC0_CK	I/O	2	PIO, I, PU, ST
							B	FLEXCOM1_IO1	I/O	2	
							C	PWML0	O	4	
							D	CANRX0	I	2	
							E	TCLK3	I	2	
C19	VDDIOP1	GPIO	PD14	I/O	–	–	A	I2SMCC0_MCK	O	2	PIO, I, PU, ST
							B	FLEXCOM1_IO2	I/O	2,3,4	
							C	PWMH0	O	4	
							D	CANTX1	O	2	
							E	TIOA4	I/O	2	
							F	FLEXCOM2_IO0	I/O	5	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
R12	VDDIOP1	GPIO	PD15	I/O	–	–	A	I2SMCC0_WS	I/O	2	PIO, I, PU, ST
							B	FLEXCOM1_IO3	I/O	2,3,4	
							C	PWML1	O	4	
							D	CANRX1	I	2	
							E	TIOB4	I/O	2	
							F	FLEXCOM2_IO1	I/O	5	
D18	VDDIOP1	GPIO	PD16	I/O	–	–	A	I2SMCC0_DOUT0	O	2	PIO, I, PU, ST
							B	FLEXCOM1_IO4	I/O	2,3,4	
							C	PWMH1	O	4	
							D	CANTX2	O	2	
							E	TCLK4	I	2	
							F	FLEXCOM3_IO0	I/O	5	
G18	VDDIOP1	GPIO	PD17	I/O	–	–	A	I2SMCC0_DOUT1	O	2	PIO, I, PU, ST
							B	FLEXCOM2_IO0	I/O	2	
							C	PWML2	O	4	
							D	CANRX2	I	2	
							E	TIOA5	I/O	2	
							F	FLEXCOM3_IO1	I/O	5	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary Signal	Primary Dir	Alternate Signal	Alternate Dir	PIO Peripheral Func	PIO Peripheral Signal	PIO Peripheral Dir	PIO Peripheral I/O Set	Reset State Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B19	VDDIOP1	GPIO	PD18	I/O	–	–	A	I2SMCC0_DOUT2	O	2	PIO, I, PU, ST
							B	FLEXCOM2_IO1	I/O	2	
							C	PWMH2	O	4	
							D	CANTX3	O	2	
							E	TIOB5	I/O	2	
							F	FLEXCOM4_IO0	I/O	5	
N11	VDDIOP1	GPIO	PD19	I/O	–	–	A	I2SMCC0_DOUT3	O	2	PIO, I, PU, ST
							B	FLEXCOM2_IO2	I/O	2,3,4,5	
							C	PWML3	O	4	
							D	CANRX3	I	2	
							E	TCLK5	I	2	
							F	FLEXCOM4_IO1	I/O	5	
C18	VDDIOP1	GPIO	PD20	I/O	–	–	A	PCK0	O	3	PIO, I, PU, ST
							B	FLEXCOM2_IO3	I/O	2,3,4,5	
							C	PWMH3	O	4	
							D	CANTX4	O	2	
							F	FLEXCOM5_IO0	I/O	5	
F18	VDDIOP1	GPIO	PD21	I/O	–	–	A	PCK1	O	3	PIO, I, PU, ST
							B	FLEXCOM2_IO4	I/O	2,3,4,5	
							D	CANRX4	I	2	
							F	FLEXCOM5_IO1	I/O	5	
							G	G1_TXEN	O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A19	VDDIOP1	GPIO	PD22	I/O	–	–	A	PDMC0_CLK	O	2	PIO, I, PU, ST
							C	PWMEXTRG0	I	4	
							D	RD1	I	2	
							F	CANTX5	O	2	
							G	G1_TX0	O	1	
K12	VDDIOP1	GPIO	PD23	I/O	–	–	A	PDMC0_DS0	I	2	PIO, I, PU, ST
							C	PWMEXTRG1	I	4	
							D	RF1	I/O	2	
							E	ISC_MCK	O	2	
							F	CANRX5	I	2	
							G	G1_TX1	O	1	
D17	VDDIOP1	GPIO	PD24	I/O	–	–	A	PDMC0_DS1	I	2	PIO, I, PU, ST
							C	PWMFI0	I	4	
							D	RK1	I/O	2	
							E	ISC_D0	I	2	
							G	G1_RXDV	I	1	
K10	VDDIOP1	GPIO	PD25	I/O	–	–	A	PDMC1_CLK	O	2	PIO, I, PU, ST
							B	FLEXCOM5_IO0	I/O	2	
							C	PWMFI1	I	4	
							D	TD1	O	2	
							E	ISC_D1	I	2	
							G	G1_RX0	I	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
G14	VDDIOP1	GPIO	PD26	I/O	–	–	A	PDMC1_DS0	I	2	PIO, I, PU, ST
							B	FLEXCOM5_IO1	I/O	2	
							C	ADTRG	I	3	
							D	TF1	I/O	2	
							E	ISC_D2	I	2	
							G	G1_RX1	I	1	
D16	VDDIOP1	GPIO	PD27	I/O	–	–	A	PDMC1_DS1	I	2	PIO, I, PU, ST
							B	FLEXCOM5_IO2	I/O	2,3,4,5	
							C	TIOA0	I/O	3	
							D	TK1	I/O	2	
							E	ISC_D3	I	2	
							G	G1_RXER	I	1	
J12	VDDIOP1	GPIO	PD28	I/O	–	–	A	RD0	I	2	PIO, I, PU, ST
							B	FLEXCOM5_IO3	I/O	2,3,4,5	
							C	TIOB0	I/O	3	
							D	I2SMCC1_CK	I/O	2	
							E	ISC_D4	I	2	
							F	PWML3	O	5	
							G	G1_MDC	O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B18	VDDIOP1	GPIO	PD29	I/O	–	–	A	RF0	I/O	2	PIO, I, PU, ST
							B	FLEXCOM5_IO4	I/O	2,3,4,5	
							C	TCLK0	I	3	
							D	I2SMCC1_WS	I/O	2	
							E	ISC_D5	I	2	
							F	PWMH3	O	5	
							G	G1_MDIO	I/O	1	
H13	VDDIOP1	GPIO	PD30	I/O	–	–	A	RK0	I/O	2	PIO, I, PU, ST
							B	FLEXCOM6_IO0	I/O	2	
							C	TIOA1	I/O	3	
							D	I2SMCC1_MCK	O	2	
							E	ISC_D6	I	2	
							F	PWMEXTRG0	I	5	
							G	G1_TXCK/ G1_REFCK	I/O	1	
C16	VDDIOP1	GPIO	PD31	I/O	–	–	A	TD0	O	2	PIO, I, PU, ST
							B	FLEXCOM6_IO1	I/O	2	
							C	TIOB1	I/O	3	
							D	I2SMCC1_DOUT0	O	2	
							E	ISC_D7	I	2	
							F	PWMEXTRG1	I	5	
							G	G1_TX2	O	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
E18	VDDIOP1	GPIO	PE0	I/O	–	–	A	TF0	I/O	2	PIO, I, PU, ST
							B	FLEXCOM6_IO2	I/O	2,3,4,5	
							C	TCLK1	I	3	
							D	I2SMCC1_DOUT1	O	2	
							E	ISC_HSYNC	I	2	
							F	PWMFI0	I	5	
							G	G1_TX3	O	1	
F14	VDDIOP1	GPIO	PE1	I/O	–	–	A	TK0	I/O	2	PIO, I, PU, ST
							B	FLEXCOM6_IO3	I/O	2,3,4,5	
							C	TIOA2	I/O	3	
							D	I2SMCC1_DOUT2	O	2	
							E	ISC_VSYNC	I	2	
							F	PWMFI1	I	5	
							G	G1_RX2	I	1	
A18	VDDIOP1	GPIO	PE2	I/O	–	–	A	PWML0	O	5	PIO, I, PU, ST
							B	FLEXCOM6_IO4	I/O	2,3,4,5	
							C	TIOB2	I/O	3	
							D	I2SMCC1_DOUT3	O	2	
							E	ISC_FIELD	I	2	
							G	G1_RX3	I	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
C15	VDDIOP1	GPIO	PE3	I/O	–	–	A	PWMH0	O	5	PIO, I, PU, ST
							B	FLEXCOM0_IO0	I/O	4	
							C	TCLK2	I	3	
							D	I2SMCC1_DIN0	I	2	
							E	ISC_PCK	I	2	
							G	G1_RXCK	I	1	
D15	VDDIOP1	GPIO	PE4	I/O	–	–	A	PWML1	O	5	PIO, I, PU, ST
							B	FLEXCOM0_IO1	I/O	4	
							C	TIOA3	I/O	3	
							D	I2SMCC1_DIN1	I	2	
							E	ISC_D8	I	2	
							G	G1_TXER	O	1	
B16	VDDIOP1	GPIO	PE5	I/O	–	–	A	PWMH1	O	5	PIO, I, PU, ST
							B	FLEXCOM0_IO2	I/O	4	
							C	TIOB3	I/O	3	
							D	I2SMCC1_DIN2	I	2	
							E	ISC_D9	I	2	
							G	G1_COL	I	1	

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
G13	VDDIOP1	GPIO	PE6	I/O	–	–	A	PWML2	O	5	PIO, I, PU, ST
							B	FLEXCOM0_IO3	I/O	4	
							C	TCLK3	I	3	
							D	I2SMCC1_DIN3	I	2	
							E	ISC_D10	I	2	
							G	G1_CRS	I	1	
E14	VDDIOP1	GPIO	PE7	I/O	–	–	A	PWMH2	O	5	PIO, I, PU, ST
							B	FLEXCOM0_IO4	I/O	4	
							C	TIOA4	I/O	3	
							E	ISC_D11	I	2	
							G	G1_TSUCOMP	O	1	
G6	VDDANA	analog input	ADVREFP	I	–	–	–	–	–	–	–
H3	VDDIN33	power	VDDIN33	I	–	–	–	–	–	–	–
H1	GNDIN33	ground	GNDIN33	I	–	–	–	–	–	–	–
F5	GNDANA	ground	GNDANA	I	–	–	–	–	–	–	–
G5	GNDANA	ground	GNDANA	I	–	–	–	–	–	–	–
J4	GNDANA	ground	GNDANA	I	–	–	–	–	–	–	–
T13	VDDIOP0	power	VDDIOP0	I	–	–	–	–	–	–	–
W14	VDDIOP0	power	VDDIOP0	I	–	–	–	–	–	–	–
C14	VDDIOP1	power	VDDIOP1	I	–	–	–	–	–	–	–
C17	VDDIOP1	power	VDDIOP1	I	–	–	–	–	–	–	–
A11	GND	ground	GND	I	–	–	–	–	–	–	–
A14	GND	ground	GND	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A17	GND	ground	GND	I	–	–	–	–	–	–	–
A21	GND	ground	GND	I	–	–	–	–	–	–	–
AA1	GND	ground	GND	I	–	–	–	–	–	–	–
W5	GND	ground	GND	I	–	–	–	–	–	–	–
AA8	GND	ground	GND	I	–	–	–	–	–	–	–
AA11	GND	ground	GND	I	–	–	–	–	–	–	–
AA14	GND	ground	GND	I	–	–	–	–	–	–	–
AA17	GND	ground	GND	I	–	–	–	–	–	–	–
AA21	GND	ground	GND	I	–	–	–	–	–	–	–
D9	GND	ground	GND	I	–	–	–	–	–	–	–
E5	VDDANA	power	VDDANA	I	–	–	–	–	–	–	–
E6	VDDANA	power	VDDANA	I	–	–	–	–	–	–	–
J6	VDDIN33	analog output	VDDOUT25	O	–	–	–	–	–	–	–
F6	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
F9	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
F15	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
J9	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
J13	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
L3	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
L19	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
N13	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
P19	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
U16	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
U17	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
U19	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
W17	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
Y20	VDDCORE	power	VDDCORE	I	–	–	–	–	–	–	–
B20	VDDCPU	power	VDDCPU	I	–	–	–	–	–	–	–
E16	VDDCPU	power	VDDCPU	I	–	–	–	–	–	–	–
E17	VDDCPU	power	VDDCPU	I	–	–	–	–	–	–	–
E19	VDDCPU	power	VDDCPU	I	–	–	–	–	–	–	–
F16	VDDCPU	power	VDDCPU	I	–	–	–	–	–	–	–
D13	GND	ground	GND	I	–	–	–	–	–	–	–
E7	GND	ground	GND	I	–	–	–	–	–	–	–
E15	GND	ground	GND	I	–	–	–	–	–	–	–
E21	GND	ground	GND	I	–	–	–	–	–	–	–
F17	GND	ground	GND	I	–	–	–	–	–	–	–
G11	GND	ground	GND	I	–	–	–	–	–	–	–
G15	GND	ground	GND	I	–	–	–	–	–	–	–
G17	GND	ground	GND	I	–	–	–	–	–	–	–
H8	GND	ground	GND	I	–	–	–	–	–	–	–
H14	GND	ground	GND	I	–	–	–	–	–	–	–
H21	GND	ground	GND	I	–	–	–	–	–	–	–
J18	GND	ground	GND	I	–	–	–	–	–	–	–
L1	GND	ground	GND	I	–	–	–	–	–	–	–
L7	GND	ground	GND	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
L11	GND	ground	GND	I	–	–	–	–	–	–	–
L15	GND	ground	GND	I	–	–	–	–	–	–	–
L21	GND	ground	GND	I	–	–	–	–	–	–	–
N4	GND	ground	GND	I	–	–	–	–	–	–	–
A1	GNDUTMI	ground	GNDUTMI	I	–	–	–	–	–	–	–
A5	GNDUTMI	ground	GNDUTMI	I	–	–	–	–	–	–	–
A8	GNDUTMI	ground	GNDUTMI	I	–	–	–	–	–	–	–
E1	GNDUTMI	ground	GNDUTMI	I	–	–	–	–	–	–	–
G7	GNDUTMI	ground	GNDUTMI	I	–	–	–	–	–	–	–
B2	VDDUTMII	power	VDDUTMII	I	–	–	–	–	–	–	–
C5	VDDUTMII	power	VDDUTMII	I	–	–	–	–	–	–	–
C8	VDDUTMII	power	VDDUTMII	I	–	–	–	–	–	–	–
E3	VDDUTMII	power	VDDUTMII	I	–	–	–	–	–	–	–
F7	VDDUTMII	power	VDDUTMII	I	–	–	–	–	–	–	–
A3	VDDUTMII	–	HHSA_DP	I/O	–	–	–	–	–	–	–
B3	VDDUTMII	–	HHSA_DM	I/O	–	–	–	–	–	–	–
D7	VDDUTMII	–	HHSA_CC1	I/O	–	–	–	–	–	–	–
G9	VDDUTMII	–	HHSA_CC2	I/O	–	–	–	–	–	–	–
C6	VDDUTMII	analog input	HHSA_RTUNE	I	–	–	–	–	–	–	–
A4	VDDUTMII	–	HHSB_DP	I/O	–	–	–	–	–	–	–
B4	VDDUTMII	–	HHSB_DM	I/O	–	–	–	–	–	–	–
C7	VDDUTMII	–	HHSB_CC1	I/O	–	–	–	–	–	–	–
E8	VDDUTMII	–	HHSB_CC2	I/O	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
D8	VDDUTMII	analog input	HHSB_RTUNE	I	–	–	–	–	–	–	–
A6	VDDUTMII	–	HHSC_DP	I/O	–	–	–	–	–	–	–
B6	VDDUTMII	–	HHSC_DM	I/O	–	–	–	–	–	–	–
B7	VDDUTMII	analog input	HHSC_RTUNE	I	–	–	–	–	–	–	–
N6	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
N9	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
N12	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
P3	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
P13	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
T6	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
T7	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
T9	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
U3	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
U5	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
U6	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
AA5	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
W8	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
Y2	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
M11	VDDIODDR	power	VDDIODDR	I	–	–	–	–	–	–	–
K11	GND	ground	GND	I	–	–	–	–	–	–	–
N18	GND	ground	GND	I	–	–	–	–	–	–	–
P1	GND	ground	GND	I	–	–	–	–	–	–	–
P8	GND	ground	GND	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
P14	GND	ground	GND	I	–	–	–	–	–	–	–
P21	GND	ground	GND	I	–	–	–	–	–	–	–
R5	GND	ground	GND	I	–	–	–	–	–	–	–
R7	GND	ground	GND	I	–	–	–	–	–	–	–
R17	GND	ground	GND	I	–	–	–	–	–	–	–
T5	GND	ground	GND	I	–	–	–	–	–	–	–
T17	GND	ground	GND	I	–	–	–	–	–	–	–
U1	GND	ground	GND	I	–	–	–	–	–	–	–
U7	GND	ground	GND	I	–	–	–	–	–	–	–
U21	GND	ground	GND	I	–	–	–	–	–	–	–
V9	GND	ground	GND	I	–	–	–	–	–	–	–
Y1	VDDIODDR	analog input	DDR_VREF	I	–	–	–	–	–	–	–
N2	VDDIODDR	–	DDR_D0	I/O	–	–	–	–	–	–	–
M7	VDDIODDR	–	DDR_D1	I/O	–	–	–	–	–	–	–
R3	VDDIODDR	–	DDR_D2	I/O	–	–	–	–	–	–	–
M8	VDDIODDR	–	DDR_D3	I/O	–	–	–	–	–	–	–
T2	VDDIODDR	–	DDR_D4	I/O	–	–	–	–	–	–	–
N8	VDDIODDR	–	DDR_D5	I/O	–	–	–	–	–	–	–
T1	VDDIODDR	–	DDR_D6	I/O	–	–	–	–	–	–	–
N7	VDDIODDR	–	DDR_D7	I/O	–	–	–	–	–	–	–
Y7	VDDIODDR	–	DDR_D8	I/O	–	–	–	–	–	–	–
AA4	VDDIODDR	–	DDR_D9	I/O	–	–	–	–	–	–	–
Y9	VDDIODDR	–	DDR_D10	I/O	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
Y3	VDDIODDR	–	DDR_D11	I/O	–	–	–	–	–	–	–
AA7	VDDIODDR	–	DDR_D12	I/O	–	–	–	–	–	–	–
AA3	VDDIODDR	–	DDR_D13	I/O	–	–	–	–	–	–	–
W7	VDDIODDR	–	DDR_D14	I/O	–	–	–	–	–	–	–
W3	VDDIODDR	–	DDR_D15	I/O	–	–	–	–	–	–	–
W6	VDDIODDR	–	DDR_A0	O	–	–	–	–	–	–	–
P6	VDDIODDR	–	DDR_A1	O	–	–	–	–	–	–	–
V5	VDDIODDR	–	DDR_A2	O	–	–	–	–	–	–	–
V6	VDDIODDR	–	DDR_A3	O	–	–	–	–	–	–	–
R6	VDDIODDR	–	DDR_A4	O	–	–	–	–	–	–	–
W4	VDDIODDR	–	DDR_A5	O	–	–	–	–	–	–	–
P7	VDDIODDR	–	DDR_A6	O	–	–	–	–	–	–	–
W2	VDDIODDR	–	DDR_A7	O	–	–	–	–	–	–	–
L8	VDDIODDR	–	DDR_A8	O	–	–	–	–	–	–	–
V4	VDDIODDR	–	DDR_A9	O	–	–	–	–	–	–	–
T3	VDDIODDR	–	DDR_A10	O	–	–	–	–	–	–	–
M9	VDDIODDR	–	DDR_A11	O	–	–	–	–	–	–	–
P5	VDDIODDR	–	DDR_A12	O	–	–	–	–	–	–	–
V2	VDDIODDR	–	DDR_A13	O	–	–	–	–	–	–	–
L9	VDDIODDR	–	DDR_A14	O	–	–	–	–	–	–	–
R4	VDDIODDR	–	DDR_A15	O	–	–	–	–	–	–	–
U4	VDDIODDR	–	DDR_CLK	O	–	–	–	–	–	–	–
T4	VDDIODDR	–	DDR_CLKN	O	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
V3	VDDIODDR	–	DDR_CKE	O	–	–	–	–	–	–	–
V1	VDDIODDR	–	DDR_RESETN	O	–	–	–	–	–	–	–
V7	VDDIODDR	–	DDR_CSN	O	–	–	–	–	–	–	–
N10	VDDIODDR	–	DDR_WEN	O	–	–	–	–	–	–	–
U8	VDDIODDR	–	DDR_RASN	O	–	–	–	–	–	–	–
T8	VDDIODDR	–	DDR_CASN	O	–	–	–	–	–	–	–
N1	VDDIODDR	–	DDR_DQM0	O	–	–	–	–	–	–	–
Y4	VDDIODDR	–	DDR_DQM1	O	–	–	–	–	–	–	–
R1	VDDIODDR	–	DDR_DQS0	I/O	–	–	–	–	–	–	–
Y6	VDDIODDR	–	DDR_DQS1	I/O	–	–	–	–	–	–	–
R2	VDDIODDR	–	DDR_DQSN0	I/O	–	–	–	–	–	–	–
AA6	VDDIODDR	–	DDR_DQSN1	I/O	–	–	–	–	–	–	–
V8	VDDIODDR	–	DDR_BA0	O	–	–	–	–	–	–	–
P4	VDDIODDR	–	DDR_BA1	O	–	–	–	–	–	–	–
R8	VDDIODDR	–	DDR_BA2	O	–	–	–	–	–	–	–
W1	VDDIODDR	–	DDR_ODT	O	–	–	–	–	–	–	–
AA2	VDDIODDR	analog input	DDR_ZQ	I	–	–	–	–	–	–	–
T15	VDDDPHY	power	VDDDPHY	I	–	–	–	–	–	–	–
T16	VDDDPHY	power	VDDDPHY	I	–	–	–	–	–	–	–
R15	GNDDPHY	ground	GNDDPHY	I	–	–	–	–	–	–	–
U15	GNDDPHY	ground	GNDDPHY	I	–	–	–	–	–	–	–
N20	VDDDPHY	–	MIPI_CLKN	I	–	–	–	–	–	–	–
N21	VDDDPHY	–	MIPI_CLKP	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
R18	VDDDPHY	–	MIPI_DN0	I	–	–	–	–	–	–	–
P18	VDDDPHY	–	MIPI_DP0	I	–	–	–	–	–	–	–
R20	VDDDPHY	–	MIPI_DN1	I	–	–	–	–	–	–	–
R21	VDDDPHY	–	MIPI_DP1	I	–	–	–	–	–	–	–
R19	VDDDPHY	analog input	MIPI_REXT	I	–	–	–	–	–	–	–
J16	VDDSDMMC0	power	VDDSDMMC0	I	–	–	–	–	–	–	–
C11	VDDSDMMC1	power	VDDSDMMC1	I	–	–	–	–	–	–	–
V13	GND	ground	GND	I	–	–	–	–	–	–	–
K15	VDDSDMMC0	analog input	SDMMC0_CAL	I	–	–	–	–	–	–	–
J10	VDDSDMMC1	analog input	SDMMC1_CAL	I	–	–	–	–	–	–	–
F13	VDDSDMMC2	power	VDDSDMMC2	I	–	–	–	–	–	–	–
D14	VDDSDMMC2	analog input	SDMMC2_CAL	I	–	–	–	–	–	–	–
R11	GNDBAT	ground	GNDBAT	I	–	–	–	–	–	–	–
P11	VBAT	power	VBAT	I	–	–	–	–	–	–	–
AA19	VBAT	PIOBU	PIOBU0	I	–	–	–	–	–	–	PU(3)
P9	VBAT	PIOBU	PIOBU1	I	–	–	–	–	–	–	PU(3)
AA18	VBAT	PIOBU	PIOBU2	I	–	–	–	–	–	–	PU(3)
P10	VBAT	PIOBU	PIOBU3	I	–	–	–	–	–	–	PD(3)
J2	VDDIN33	–	XIN	I	–	–	–	–	–	–	–
J1	VDDIN33	–	XOUT	O	–	–	–	–	–	–	–
Y13	VBAT	–	XIN32	I	–	–	–	–	–	–	–
AA13	VBAT	–	XOUT32	O	–	–	–	–	–	–	–
AA9	VBAT	–	TST	I	–	–	–	–	–	–	–

...........continued

343-pin TFBGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral				Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
W15	VBAT	–	JTAGSEL	I	–	–	–	–	–	–	–
AA15	VBAT	–	WKUP0	I	–	–	–	–	–	–	–
R9	VBAT	–	SHDN	O	–	–	–	–	–	–	–
J7	VDDIN33	–	NRST	I	–	–	–	–	–	–	–
H4	VDDIN33	–	NRST_OUT	O	–	–	–	–	–	–	–
K7	VDDIN33	GPIO	AUDIOCLK	O	–	–	–	–	–	–	–
R10	VBAT	–	LPM	O	–	–	–	–	–	–	–
N16	VDDQSPI0	power	VDDQSPI0	I	–	–	–	–	–	–	–
P16	VDDQSPI0	analog input	QSPI0_CAL	I	–	–	–	–	–	–	–
H19	VDDQSPI1	power	VDDQSPI1	I	–	–	–	–	–	–	–

Notes:

1. I/Os for each peripheral are grouped into I/O sets, listed in the column "I/O Set". For all peripherals, use I/Os that belong to the same I/O set. Timings can be unpredictable when I/Os from different I/O sets are mixed.
2. When using an I/O line with the Analog-to-Digital Converter (ADC) or with the Analog Comparator Controller (ACC), the PIO line configuration (pull-up, pull-down) programmed before assigning this line to the ADC or ACC peripheral is not modified by this peripheral.
3. This is the PIOBU state after VBAT power-up. If programmed to another value, this value is maintained as long as VBAT is not removed.

8. System Interconnect and Security (SIS)

8.1 System Bus and Interconnect

The device on-chip interconnect is architectured around the following components:

- 5x AXI matrixes based on the Arm NIC-400 module
- 1x AHB matrix
- 8x APB buses
- 1x Universal DDR Memory Controller (UDDRC)

The following features are supported by the interconnect backbone:

- Quality of Service (QoS) to ensure priorities and limits for all AXI and AHB transactions
- Performance monitoring to track AXI bus activity

8.1.1 Block Diagram

Figure 8-1. SAMA7G5 Architecture

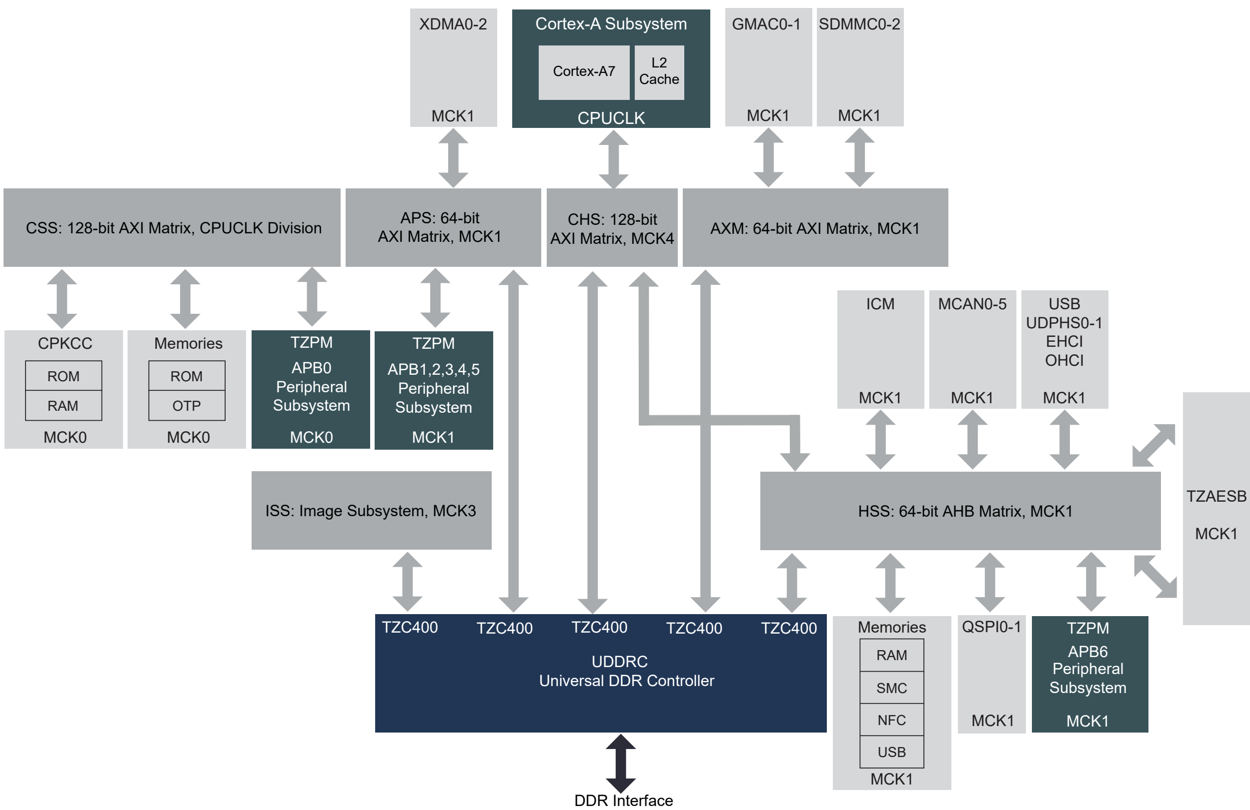


Table 8-1. Advanced Peripheral Bus User Interfaces

Advanced Peripheral Bus	User Interfaces
APB0	SECURAM, SECUMOD, SFRBU, CPKCC, TZPM, PIOA, PMC, SYSC
APB1	CSI, CSI2DC, ISC, XDMA2, SDMMC0–2, ADCC, ACC, UDDRPUBL, PWM, PDMC0–1, ASRC, SPDIFRX, SPDIFTX, I2SMCC0–1, SFR, EIC
APB2	PIT64B0–2, SSC0, AES, SHA, FLEXCOM0–3

...........continued

Advanced Peripheral Bus	User Interfaces
APB3	TZAESBASC, PIT64B3–4, SSC1, TRNG, TDES, FLEXCOM4–7
APB4	GMAC0–1, XDMAC0–1, PIT64B5, TC0, FLEXCOM8–11
APB5	TZC
APB6	MCAN0–5, TC1, MATRIX, SMC, QSPI0–1, UDPHS0–1, ICM, TZAESB, TCPCA, TCPCB
APB7	UDDRC, DBG

8.1.2 AXI Subsystem

The AXI subsystem is comprised of:

- CSS — CPU System and Security matrix
- CHS — CPU High Speed matrix
- APS — APB Client matrix
- AXM — AXI Hosts matrix
- ISS — Image Subsystem

AXI matrixes are based on NIC-400 r1p1 (by Arm Ltd.) and no settings are controlled by software. The respective hardware configurations used are given in the following sections. The default software configuration has been intensively tested and leads to best results in any conditions.

For complete details on the NIC-400 design, see the Arm specification on http://infocenter.arm.com/help/topic/com.arm.doc.ddi0475h/index.html

The device embeds Quality of Service management provided by Arm QoS-400 supplement to NIC-400. QoS-400 r1p1 is controlled by software. The respective hardware configurations used are given in the following sections. The default software configuration has been intensively tested and leads to best results in any conditions.

For complete details on the QoS-400, see the Arm specification on http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dsu0026g/index.html

This AXI subsystem (NICGPV) is connected to peripherals and to an AHB matrix (MATRIX) described below.

The following tables summarize the AXI matrix hosts and clients.

Table 8-2. AXI Hosts

Host Port	Port Name
DDRC_P0 CHS	AMIB0
ROM/OTP CSS	AMIB1
CPKCC CSS	AMIB2
APB0 CSS	AMIB3
Bridge to HSS M0 CHS	AMIB4
DDRC_P2 APS	AMIB5
Bridge to HSS M1 APS	AMIB6
APB1 APS	AMIB7
APB2 APS	AMIB8
APB3 APS	AMIB9
APB4 APS	AMIB10
APB7 APS	AMIB11

...........continued	
Host Port	**Port Name**
Bridge to HSS M2 AXM	AMIB12
DDR_P4 AXM	AMIB13
CSI2DC AXM	AMIB14

Table 8-3. AXI Clients

Client Port	Port Name
CPU CHS	ASIB0
OTP CSS	ASIB1
XDMAC0 APS	ASIB2
XDMAC1 APS	ASIB3
DEBUG APS	ASIB4
GMAC0 AXM	ASIB5
GMAC1 AXM	ASIB6
SDMMC0 AXM	ASIB7
SDMMC1 AXM	ASIB8
SDMMC2 AXM	ASIB9
XDMAC2 AXM	ASIB10

8.1.3 AHB Subsystem

The following tables summarize the AHB matrix hosts and clients.

Table 8-4. AHB Hosts

Host Port	Port Name
CPU from AXI	M0
XDMAC0–1 from AXI	M1
GMAC0–1 from AXI SDMMC0–2 from AXI XDMAC2 from AXI	M2
MCAN0	M3
MCAN1	M4
MCAN2	M5
MCAN3	M6
MCAN4	M7
MCAN5	M8
ICM	M9
UDPHS0_DMA	M10
UDPHS1_DMA	M11

...........continued	
Host Port	**Port Name**
OHCI_DMA	M12
EHCI_DMA	M13
TZAESB	M14

Table 8-5. AHB Clients

Client Port	Port Name
QSPI0	S0
QSPI1	S1
TZAESB	S2
UDDRC_P1	S3
APB6	S4
SRAM_P0	S5
SRAM_P1	S6
SMC[(1)]	S7
NFC_RAM	S8
USB_RAM	S9

Note:

1. The Static Memory Controller (SMC) contains several configurable memory areas. These are EBI_CS0, EBI_CS1, EBI_CS2, EBI_CS3 and NFC_CMD, with respective HSEL from HSEL0 to HSEL4.

To improve performance, a transaction QoS can be both generated and handled by the matrix. See Quality of Service (QoS) Overview.

8.2 System Interconnect Overview

The following table shows allowed paths (X).

Table 8-6. System Interconnections

		UDDRC_P0 – MCK4	UDDRC_P1 – MCK1	UDDRC_P2 – MCK1	UDDRC_P3 – MCK3	UDDRC_P4 – MCK1	OTPC – MCK0	CPKCC RAM/ROM – MCK0	APB0 – MCK0	APB1 – MCK1	APB2 – MCK1	APB3 – MCK1	APB4 – MXK1	APB5 – MCK1	APB6 – MCK1	APB7 – MCK1	USB_RAM – MCK1	SRAM_P0 – MCK1	SRAM_P1 – MCK1	TZAESB – MCK1	SMC – MKC1	QSPI0 – MCK1	QSPI1 – MCK1	NFC_RAM – MCK1	NFC_CMD – MCK1	UHPHS_OHCI/EHCI – MCK1	APB_DBG/APB_DBG_S – MCK1	NICGPV – MCK1	CSI2DC_META – MCK1	ROM/ ECC_ROM – MCK0
TrustZone Management Location[1]		TZC	TZC	TZC	TZC	TZC	AS[2]	AS [2]	AS[2]	TZPM	TZPM	TZPM	TZPM	TZPM	TZPM	TZPM	MATRIX	MATRIX	MATRIX	TZAESBAS -CTZPM	MATRIX	MATRIX	MATRIX	MATRIX	MATRIX	MATRIX	MATRIX	AS[2]	MCK1	MCK0
CA7 – MCK4	Supervisor mode or CP15	X	–	–	–	–	X	X	X	X	X	X	X	X	X	X	X	X	–	X	X	X	X	X	X	X	X	X	–	X
XDMAC0–MCK1	XDMAC0	–	–	X	–	–	–	–	–	X	X	X	X	–	X	–	–	–	X	X	X	X	X	X	–	–	–	–	–	–
XDMAC1–MCK1	XDMAC1	–	–	X	–	–	–	–	–	X	X	X	X	–	X	–	–	–	X	X	X	X	X	X	–	–	–	–	–	–
GMAC0–MCK1	TZPM	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	–	X	X	X	–	–	–	–	–	–	–
GMAC1–MCK1	TZPM	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	–	X	X	X	–	–	–	–	–	–	–
SDMMC0–MCK1	TZPM	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	–	X	X	X	–	–	–	–	–	–	–
SDMMC1–MCK1	TZPM	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	–	X	X	X	–	–	–	–	–	–	–
SDMMC2–MCK1	TZPM	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	–	X	X	X	–	–	–	–	–	–	–
XDMAC2–MCK1	XDMAC2	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	X	X	X	X	X	X	–	–	–	–	X	–
MCAN0 –MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–
MCAN1–MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–
MCAN2–MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–
MCAN3–MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–
MCAN4–MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–

...........continued

		UDDRC_P0 – MCK4	UDDRC_P1 – MCK1	UDDRC_P2 – MCK1	UDDRC_P3 – MCK3	UDDRC_P4 – MCK1	OTPC – MCK0	CPKCC RAM/ROM – MCK0	APB0 – MCK0	APB1 – MCK1	APB2 – MCK1	APB3 – MCK1	APB4 – MXK1	APB5 – MCK1	APB6 – MCK1	APB7 – MCK1	USB_RAM – MCK1	SRAM_P0 – MCK1	SRAM_P1 – MCK1	TZAESB – MCK1	SMC – MKC1	QSPI0 – MCK1	QSPI1 – MCK1	NFC_RAM – MCK1	NFC_CMD – MCK1	UHPHS_OHCI/EHCI – MCK1	APB_DBG/APB_DBG_S – MCK1	NICGPV – MCK1	CSI2DC_META – MCK1	ROM/ ECC_ROM – MCK0
MCAN5–MCK1	TZPM	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–
ICM–MCK1	TZPM	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	X	X	X	–	–	–	–	–	–	–
UDPHS0_DMA–MCK1	TZPM	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–
UDPHS1_DMA–MCK1	TZPM	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–
OHCI_DMA – MCK1	TZPM	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–
EHCI_DMA–MCK1	TZPM	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–
TZAESB–MCK1	TZAESB ASC	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	X	X	X	X	–	–	–	–	–	–
ISC–MCK3	TZPM	–	–	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Notes:
1. Refer to the following sections for details on each configuration method.
2. “AS” stands for Always Secure; this configuration cannot be changed.

8.3 Quality of Service (QoS) Overview

Quality of Service is extremely important in distributing the available resources (bandwidth or latency) among devices to meet overall performance requirements.

In the circuit, multiple hosts communicate with clients through interconnect. However, bandwidth-sensitive hosts can starve for data and latency-sensitive host requirements can be unsatisfied. This impacts the overall performance of the subsystem. QoS and Outstanding Capability address these types of performance issues in the subsystem.

Configuration can be done in different locations: NIC-400 Global Programmer's View (NICGPV), MATRIX, peripheral, UDDR Controller or SFR register.

Some peripherals embed bandwidth regulation. This feature is located in the NICGPV.

8.3.1 Cortex-A7 CPU

The Cortex-A7 CPU does not generate its own QoS. The CPU QoS is programmable in the interconnect matrix through the NICGPV registers.

The ASIB block that generates the ARQOS and AWQOS signals is configured through register ASIB[0].

8.3.2 DMA

8.3.2.1 DMA Per Queue Outstanding Capabilities

- Each XDMA transfer stream is managed by 7 transfer queues (3 for write operations, 4 for read operations).

- The outstanding capability of each of those queues is configurable through the XDMAC_GCFG register fields.
- Each queue has a static priority versus the other queues.

Register fields:

- XDMAC_GCFG.WRHP : write outstanding for peripheral write of a memory-to-peripheral DMA channel (highest write priority)
- XDMAC_GCFG.WRMP: write outstanding for memory write of a peripheral-to-memory DMA channel (medium write priority)
- XDMAC_GCFG.WRLP: write outstanding for memory write of a memory-to-memory DMA channel (lowest write priority)
- XDMAC_GCFG.RDHP: read outstanding for peripheral read of a peripheral-to-memory DMA channel (high read priority)
- XDMAC_GCFG.RDMP: read outstanding for memory read of a memory-to-peripheral DMA channel (medium read priority)
- XDMAC_GCFG.RDSP: read outstanding for transfer descriptor read of all types of DMA channel (highest priority)

8.3.2.2 DMA Per Queue Outstanding Capabilities

- Each channel QoS is configured through a 2-bit wide register: XDMAC_CNDC[channel number].qos.
- The 2-bit XDMAC_CNDC[channel number].qos value is duplicated to form a 4-bit bus that connects XDMAs to the UDDRC through the interconnect, therefore not all 4-bit QoS values are available.
  Possible values = {0, 0x5, 0xA, 0xF}.
- The XDMAC_CNDC[channel number].qos is only used by the following DMA transactions:
  – Read New Descriptor
  – Write Last Burst of Data before Reading New Descriptor
- For all other DMA transfers, the 4-bit QOS is 0 (lowest priority).

Register fields:

- XDMAC_CNDC[channel_number].qos = 3 for channel_number in [31:0] for XDMA0
- XDMAC_CNDC[channel_number].qos = 3 for channel_number in [31:0] for XDMA1
- XDMAC_CNDC[channel_number].qos = 0 for channel_number in [7:0] for XDMA2

8.3.3 GMAC

8.3.3.1 GMAC Outstanding Capabilities

GMAC0-1 are implemented with maximum hardware outstanding capability of four AXI read transactions and four AXI write transactions.

Register field:

The GMAC_AMP (GMAC AXI Max Pipeline) register is used to set the real outstanding capability for each GMAC (lower than the maximum implemented capability).

8.3.3.2 GMAC QoS

GMAC does not generate its own QoS. GMAC QoS is programmable in the interconnect matrix through the NICGPV registers.

The ASIB block that generates the ARQOS and AWQOS signals is configured through register ASIB[5] for GMAC0 and ASIB[6] for GMAC1.

8.3.4 SDMMC

SDMMC provides a QoS programming for AHB accesses when fetching descriptors in SDMMC0_ACR register. For all other accesses, the QOS is set to 0.

8.3.5 ISC

ISC provides dynamic (host-generated) QoS information in ISC_DCFG register. If AWQOS is set to 0, the QoS value depends on the output FIFO level (dynamic configuration). Otherwise, the value is defined in the register. ARQOS[3:0] returns the QoS bus value when a descriptor is retrieved from the memory.

8.3.6 AHB

The AHB matrix (MATRIX) provides two mechanisms:

- One inside MATRIX for each port, with registers MATRIX_PRAS[port number], MATRIX_PRBS[port number]
- One for the AHB to AXI bridge, with register SFR_HSS_AXIQOS

8.3.7 UDDRC

The five ports are configured with PCFGQOS0_[port number], PCFGQOS1_[port number], PCFGWQOS0_[port number], PCFGWQOS1_[port number].

The following table summarizes how to program Outstanding Capability in each applicable section.

Table 8-7. Outstanding Capability Programming

Instance	AXI Outstanding Management
DMA0	XDMAC0_GCFG.WRHP, XDMAC0_GCFG.WRMP, XDMAC0_GCFG.WRLP, XDMAC0_GCFG.RDHP, XDMAC0_GCFG.RDMP, XDMAC0_GCFG.RDSP
DMA1	XDMAC1_GCFG.WRHP, XDMAC1_GCFG.WRMP, XDMAC1_GCFG.WRLP, XDMAC1_GCFG.RDHP, XDMAC1_GCFG.RDMP, XDMAC1_GCFG.RDSP
DMA2	XDMAC2_GCFG.WRHP, XDMAC2_GCFG.WRMP, XDMAC2_GCFG.WRLP, XDMAC2_GCFG.RDHP, XDMAC2_GCFG.RDMP, XDMAC2_GCFG.RDSP
GMAC0	GMAC0_AMP register
GMAC1	GMAC1_AMP register

The following table summarizes how to program QoS in each applicable section.

Table 8-8. Quality of Service Programming

Instance	QoS Management	Default Value	Transaction Outstanding Latency Regulation in NICGPV
CPU	ASIB[0] in NICGPV[(1)]	0	–
DMA0, per channel	QoS in XDMAC0_CNDC[channel number] registers	0	–
DMA1, per channel	QoS in XDMAC1_CNDC[channel number] registers	0	–
DMA2, per channel	QoS in XDMAC2_CNDC[channel number] registers	0	X
GMAC0	ASIB[5] in NICGPV	0	X
GMAC1	ASIB[6] in NICGPV	0	X
SDMMC0	Peripheral for descriptors, in SDMMC0_ACR register	0	X
SDMMC1	Peripheral for descriptors, in SDMMC1_ACR register	0	X
SDMMC2	Peripheral for descriptors, in SDMMC2_ACR register	0	X
ISC	Peripheral in ISC_DCFG register	0	–
AHB matrix port 3	SFR_HSS_AXIQOS Register	0	–
DDR port, per port	UDDRC in PCFGQOS0_[port number], PCFGQOS1_[port number], PCFGWQOS0_[port number], PCFGWQOS1_[port number] registers	0	–

...........continued			
Instance	QoS Management	Default Value	Transaction Outstanding Latency Regulation in NICGPV
AHB matrix hosts (including CPU)	MATRIX_PRAS[port number] MATRIX_PRBS[port number]	0 2	–

Note:
1. This QoS is not propagated to the AHB matrix through the M0 port.

For details, refer to the relevant peripheral sections and to 12. NIC-400 Global Programmer's View (NICGPV), 13. Bus Matrix (MATRIX) and 31. Special Function Registers (SFR).

8.4 TrustZone Security Management

The device architecture embeds several mechanisms for system TrustZone configuration:

- Core security extensions
- TrustZone Peripheral Manager (TZPM) for peripheral configuration
- TrustZone registers in the AHB matrix (MATRIX) for host configuration and memory configuration (except for DDR)
- TrustZone Address Space Controller (TZC) based on Arm TZC400 modules for DDR access configuration
- TrustZone AESB Address Space Controller (TZAESBASC) for on-the-fly encrypted memory zone access

Important: The above TrustZone management mechanisms use separate configuration interfaces. Some memories or IPs may be configured by many of those mechanisms. In such case, consistency must be ensured between configurations.

8.4.1 TrustZone System Overview

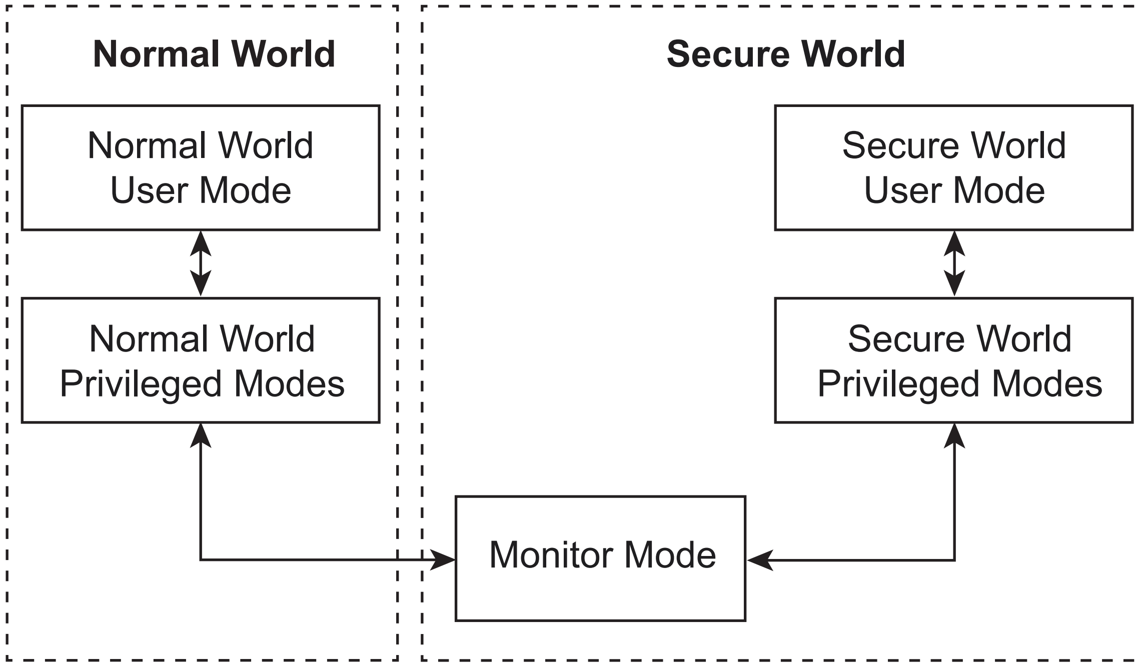
8.4.1.1 Hardware

TrustZone enables a single physical processor core to execute code safely and efficiently from both the Normal world and the Secure world. This removes the need for a dedicated security processor core, saving silicon area and power, and allowing high performance security software to run alongside the Normal world operating environment.

The two virtual processors switch contexts via Monitor mode when changing the currently running virtual processor.

See the following figure.

Figure 8-2. TrustZone Hardware Implementation



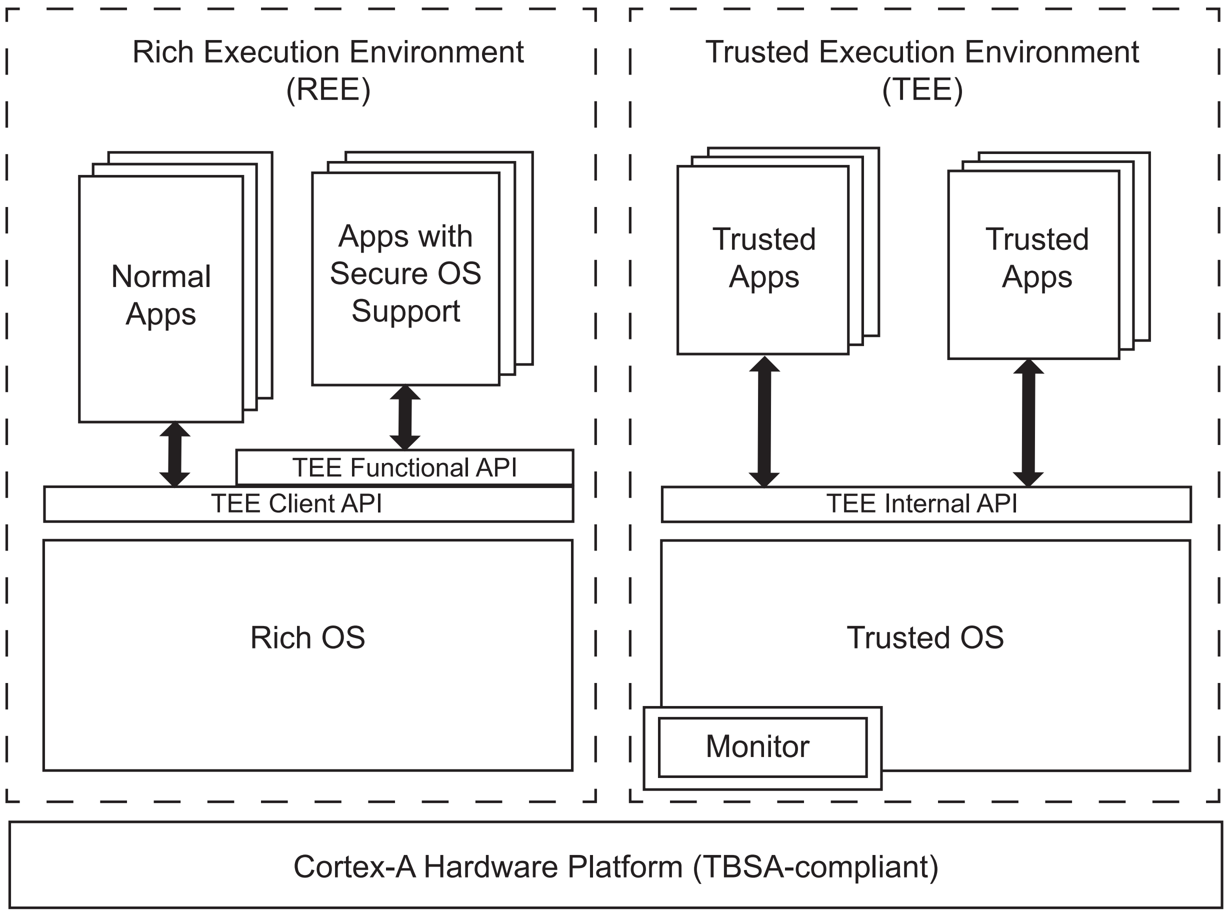
8.4.1.2 Software

The mechanisms by which the physical processor can enter Monitor mode from the Normal world are tightly controlled, and are all viewed as exceptions to the Monitor mode software. Software executing a dedicated instruction can trigger entry to monitor, the Secure Monitor Call (SMC) instruction, or by a subset of the hardware exception mechanisms. Configuration of the IRQ, external Data Abort, and external Prefetch Abort exceptions can cause the processor to switch to Monitor mode.

The software executed within Monitor mode is implementation-defined, but it generally saves the state of the current world and restores the state of the world at the location to which it switches. It then performs a return-from-exception to restart processing in the restored world.

See the following figure.

Figure 8-3. TrustZone Software Implementation in a Trusted Execution Environment (TEE)



8.4.1.3 Debug

The TrustZone hardware architecture is a security-aware debug infrastructure that can enable control over access to Secure world debug without impairing debug visibility of the Normal world. Arm CoreSight infrastructure uses four signals to control authentication:

- DBGEN – Non-secure invasive debug enable
- NIDEN – Non-secure non-invasive debug enable
- SPIDEN – Secure invasive debug enable
- SPNIDEN – Secure non-invasive debug enable

These signals allow several Debug modes and are managed in the Security Module. Refer to the section Security Module (SECUMOD) for more details.

8.4.2 Core Security Extensions Overview

The purpose of the security extensions is to enable the construction of a secure software environment.

When the Cortex-A7 processor is executing in Non-secure state, the processor performs translation table look-ups using the Non-secure versions of the translation table base registers. In this situation, any virtual address (VA) can only translate into a Non-secure physical address (PA). In Secure state, the Cortex-A7 processor performs translation table look-ups using the Secure versions of the translation table base registers. In this situation, the security state of any VA is determined by the NS bit of the translation table descriptors for that address.

Following is an example of the address manipulation that occurs when the Cortex-A7 processor requests an instruction:

1. The Cortex-A7 processor issues the instruction VA as a Secure or Non-secure VA.

2. The instruction cache is indexed by the VA bits. The MMU performs the translation table look-up in parallel with the cache access. If the processor is in the Secure state it uses the Secure translation tables, otherwise it uses the Non-secure translation tables.
3. If the protection check carried out by the MMU on the VA does not abort and the PA tag is in the instruction cache, the instruction data is returned to the processor.
4. If there is a cache miss, the MMU passes the PA to the AXI bus interface to perform an external access. The external access is always Non-secure when the core is in the Non-secure state. In the Secure state, the external access is Secure or Non-secure according to the NS attribute value in the selected translation table entry. In Secure state, both L1 and L2 translation table walk accesses are marked as Secure, even if the first level descriptor is marked as NS.

Refer to the *Arm Architecture Reference Manual, Arm v7-A and Arm v7-R edition* for details on security extensions.

8.4.3 TrustZone Peripheral Manager (TZPM)

8.4.3.1 Function

The TrustZone Peripheral Manager (TZPM) controls the access permissions to peripherals (in other words, whether or not the Non-secure world is allowed to access the peripherals). For peripherals with a bus host interface, the TZPM controls at the same time the security of host accesses transmitted by these peripherals.

The permission for each peripheral access is controlled by a dedicated security bit, the index of which is equal to the peripheral ID (unless Exceptions apply) with:

- 1: Peripheral is not secure
- 0: Peripheral is secure (cannot be accessed by Non-secure world)

128 security bits, controlling up to 128 peripherals, are organized in four registers (TZPM_PIDR0 to 3). Each register can be written if TZPM_KEY is written previously with the correct key.

- For Always Secure (AS) peripherals, the corresponding bit value is 0 and read-only.
- For Never Secure (NS) peripherals, the corresponding bit value is 1 and read-only.
- For Programmable Secure (PS) peripherals, the corresponding bit value is 0 (Secure) after reset and can be modified.

For details, see TZPM registers, Peripheral Clocks and Security and the table Peripheral Identifiers.

8.4.3.2 Example

For FLEXCOM2, the peripheral ID is ID_FLEXCOM2 = 40, requiring modification of bit TZPM_PIDR1[8].

- TZPM_KEY = 0x12AC4B5D
- TZPM_PIDR1 = 0x100

8.4.3.3 Exceptions

Not all bits are coded. In several cases, the peripheral security access is not controlled through the TZPM: Always Secure IDs, TZC400 protected IDs, specific IPs (secure DMA, OTP).

For some peripherals, the bit index controlling the access permission differs from the peripheral ID. This is the case when several peripherals share a common security bit:

- CSI, CSI2DC and ISC permissions are controlled by the bit index ID_ISC.
- DDRPUBL and DDRUMCTL permissions are controlled by the bit index ID_DDRPUBL.

Some peripheral IDs do not correspond to a peripheral user interface but only to an interrupt (ex: ID_AES_SINT). In this case, the security bit is read-only and generally equal to zero except when the associated interrupt targets only the Non-secure world. Note that the security bit value should be considered as don't-care for interrupt-only IDs.

Some peripheral IDs are not assigned to any functionality (“reserved”). The corresponding security bit is 0 (don't care), read-only.

ID_TZAESB_NS is the only programmable-secure peripheral with value 1 (not secure) after reset.

ID_UHPHS is the only programmable-secure peripheral with user interface security controlled by HMatrix programming. The corresponding TZPM bit only controls the security of USB host ports, due to the fact that the USB Host is a native AHB peripheral and is plugged directly on HMatrix (not plugged on an HBridge).

8.4.3.4 Bus Hosts

The security bits set in the TZPM control the security of bus accesses generated by the hosts. As an example, if the security bit of ID_GMAC0 is set to secure, GMAC0 will perform secure host accesses on the bus. It is not possible to have different permission levels on the user (client) interface and on the host interface. Peripherals are either only accessible by the Secure world and generate secure host accesses, or are accessible by all worlds and generate Non-secure host accesses.

Notes:

1. As any peripheral, TZPM has a dedicated peripheral ID with a corresponding security bit controlled in TZPM_PIDR registers. However, TZPM is Always Secure, so this bit is read 0, read-only.
2. Some “interrupt only” type peripheral IDs may be greater than 127 (which is the maximum ID TZPM can control). All peripherals with a user interface have an ID number below 127 in order to be controllable by TZPM.
3. The security bit value of peripheral IDs can conflict with the security level of interrupt IDs. In this case, the security level of the peripheral prevails. Remember that the security bit value information is not very important for interrupts since there is no hardware mechanism using the security bit to route the interrupts to Secure or Non-secure worlds. Interrupt security is entirely controlled by GIC programming.

Peripheral ID	Type	Security
ID_TZPM	User interface	Always Secure

8.4.3.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	TZPM_KEY	31:24	TZPMKEY[31:24]							
		23:16	TZPMKEY[23:16]							
		15:8	TZPMKEY[15:8]							
		7:0	TZPMKEY[7:0]							
0x08	TZPM_PIDR0	31:24	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
		23:16	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
		15:8	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
		7:0	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
0x0C	TZPM_PIDR1	31:24	PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
		23:16	PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
		15:8	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
		7:0	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
0x10	TZPM_PIDR2	31:24	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
		23:16	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
		15:8	PID79	PID78	PID77	PID76	PID75	PID74	PID73	PID72
		7:0	PID71	PID70	PID69	PID68	PID67	PID66	PID65	PID64
0x14	TZPM_PIDR3	31:24	PID127	PID126	PID125	PID124	PID123	PID122	PID121	PID120
		23:16	PID119	PID118	PID117	PID116	PID115	PID114	PID113	PID112
		15:8	PID111	PID110	PID109	PID108	PID107	PID106	PID105	PID104
		7:0	PID103	PID102	PID101	PID100	PID99	PID98	PID97	PID96

8.4.3.5.1 TrustZone Peripheral Manager Key

Name: TZPM_KEY
Offset: 0x0004
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TZPMKEY[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TZPMKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TZPMKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TZPMKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TZPMKEY[31:0] Key preventing TZPM_PIDx registers to be modified unintentionally by software

Value	Description
`0x12AC4B5D`	TZPM_PIDx register write is possible.
`Other values`	TZPM_PIDx register write is impossible.

8.4.3.5.2 TrustZone Peripheral Manager Peripheral ID Register 0

Name: TZPM_PIDR0
Offset: 0x0008
Reset: 0x0000F002
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral ID x Security

Value	Description
0	Secure accesses only.
1	Non-secure accesses are allowed.

8.4.3.5.3 TrustZone Peripheral Manager Peripheral ID Register 1

Name: TZPM_PIDR1
Offset: 0x000C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral ID x Security

Value	Description
0	Secure accesses only.
1	Non-secure accesses are allowed.

8.4.3.5.4 TrustZone Peripheral Manager Peripheral ID Register 2

Name: TZPM_PIDR2
Offset: 0x0010
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID79	PID78	PID77	PID76	PID75	PID74	PID73	PID72
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID71	PID70	PID69	PID68	PID67	PID66	PID65	PID64
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral ID x Security

Value	Description
0	Secure accesses only.
1	Non-secure accesses are allowed.

8.4.3.5.5 TrustZone Peripheral Manager Peripheral ID Register 3

Name: TZPM_PIDR3
Offset: 0x0014
Reset: 0x00000004
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PID127	PID126	PID125	PID124	PID123	PID122	PID121	PID120
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID119	PID118	PID117	PID116	PID115	PID114	PID113	PID112
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID111	PID110	PID109	PID108	PID107	PID106	PID105	PID104
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID103	PID102	PID101	PID100	PID99	PID98	PID97	PID96
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral ID x Security

Value	Description
0	Secure accesses only.
1	Non-secure accesses are allowed.

8.4.4 AHB Matrix (MATRIX)

8.4.4.1 Function

The AHB matrix defines security regions in the memory space. It also retrieves information after access denial events through interrupts. MATRIX covers the memory space not covered by TZPM and targets internal or external memories, except for DDR.

Peripheral ID	Type	Security
ID_MATRIX	User interface	Always Secure

8.4.4.2 Programming

Depending on its configuration and current request, the MATRIX grants or denies access to the memory areas. The Static Memory Controller (SMC), connected on S7 port, contains several configurable memory areas. These are EBI_CS0, EBI_CS1, EBI_CS2, EBI_CS3 and NFC_CMD, with respective HSEL from HSEL0 to HSEL4. For internal memory NFC_CMD, the top address is known so only MATRIX Security Areas Split Client Register 7 (MATRIX_SASSR7) is to be programmed.

Refer to the section “Bus Matrix (MATRIX)”.

8.4.5 TrustZone Address Space Controller (TZC400)

The TrustZone Controller is based on TZC400 r0p1 (by Arm Ltd.) and is highly configurable at the hardware design stage. Several options are controlled by software. The respective hardware configurations used are given in table

TZC Controller and DDR Ports. The default software configuration has been intensively tested and leads to best results in any conditions.

For complete details on the TZC400 design, refer to www.arm.com/products/silicon-ip-security/tzc-400.

8.4.5.1 Function

The TZC400 is dedicated to DDR security settings. As the DDR controller features five client ports and the maximum number of ports on a TZC400 is four, two TZC400 controllers are integrated. A 1-port TZC400 is used between the processor and the DDR, while the other four ports are secured through a 4-port TZC400. See the following table for the association of TZC ports and DDR ports.

Table 8-9. TZC Controller and DDR Ports

TZC Controller	TZC Port	DDR Port
0	0	Port 1 (TZAESB, QSPI, USB, EBI, ICM)
0	1	Port 2 (DMA0, DMA1)
0	2	Port 3 (ISC)
0	3	Port 4 (GMAC0-1, SDMMC0-2, DMA2)
1	0	Port 0 (CPU)

8.4.5.2 Filters

On the TZC400, each port has its own dedicated filter. Each filter must be configured by software; up to nine regions can be associated to a filter. A region is a memory space with particular access permissions.

TZC0 and TZC1 regions must be programmed in a consistent way. For each region, program Region ID Access register as follows:

- Set bit 0 (NSAID_RD_EN field) to allow NS read accesses. The bit default value is 0 and only S read accesses are allowed.
- Set bit 16 (NSAID_WD_EN field) to allow NS write accesses. The bit default value is 0 and only S write accesses are allowed.

Region 0 is the background region and defines the two extremities of the accessible memory range and the default access permission. This region is not optional. Accessing an address out of region 0 is always denied.

Up to eight additional regions can be defined and superimposed on region 0 but must never overlap each other. Each region can describe different access permissions (read enabled, write enabled, secure access rights, not secure access rights).

Finally, different regions can be associated to different ports (or, simply, the same regions can be associated to all ports).

In case of access permission violation, an interrupt can be generated, and denied access information (address, port, type, etc.) can be retrieved in Status registers. An interrupt can also be issued in case of programming errors (overlapping regions, for example).

Note:
The two TZC400 controllers have been merged into a combined 5-port TZC400. There is only one interrupt line, comprised of ORed interrupts of the two controllers.

After reset, the TZC400 blocks all accesses to DDR. It must be configured before any DDR access. The DDRC clock must be enabled before configuring TZC400.

Speculative access is forbidden in order to prevent the TZC400 from propagating accesses even when they fail the security permissions (this would not result in any successful access, but would lead to some visible activity on DDR ports). This restriction implies that the 'fast path' feature cannot be used in TZC400 (this feature offers more bandwidth but requires setting Speculative Access mode).

Peripheral ID	Type	Security
ID_TZC	User interface	Always Secure

8.4.6 Programmable IO Controller (PIOC)

The PIO controller memory map is split in two parts: the lower 4 Kbytes are never secure, while the upper part is always secure. Each IO line can be configured as secure or non-secure. Once configured as secure in the secure part registers, only the Secure world can reprogram the corresponding IO. It is not possible to reconfigure a secure IO by modifying the non-secure part registers as long as this IO is set as secure.

Interrupts related to events on secure IOs are transmitted on secure lines (ID_PIO*_SINT). Non-secure IO events are transmitted on non-secure interrupt lines (ID_PIO*). The GIC must be configured in order to associate the secure interrupt lines to the secure interrupts group.

Table 8-10. PIO Controller Security Settings

Peripheral ID	Type	Security
ID_PIOA	User interface interrupt ID	Secure except the first 4 Kbytes (Non-secure). The interrupt targets Non-secure world [(1)].
ID_PIOB	Interrupt ID only	Non-secure
ID_PIOC	Interrupt ID only	Non-secure
ID_PIOD	Interrupt ID only	Non-secure
ID_PIOE	Interrupt ID only	Non-secure
ID_PIOA_SINT	Interrupt ID only	Secure
ID_PIOB_SINT	Interrupt ID only	Secure
ID_PIOC_SINT	Interrupt ID only	Secure
ID_PIOD_SINT	Interrupt ID only	Secure
ID_PIOE_SINT	Interrupt ID only	Secure

Note: The ID_PIOA security bit value is 'secure', whereas the corresponding interrupt ID_PIOA is dedicated to the Non-secure world. ID_PIOA must be secure so that the PIO memory map is secure above 4 Kbytes (a mechanism implemented in HBridge). The security bit value of peripheral IDs can conflict with the security level of interrupt IDs. In such case, the peripheral security level always prevails.

Refer to the section Parallel Input/Output Controller (PIO) for more details.

8.4.7 TrustZone AES Bridge Address Space Controller (TZAESBASC)

8.4.7.1 Function

The TrustZone AES Bridge (TZAESB) performs encrypted memory accesses. It also offers the possibility for Secure and Non-secure worlds to perform encrypted memory accesses with different encryption keys. The memory space that can be encrypted by TZAESB include the NFC_RAM, EBI, SRAM, QSPI and DDR spaces.

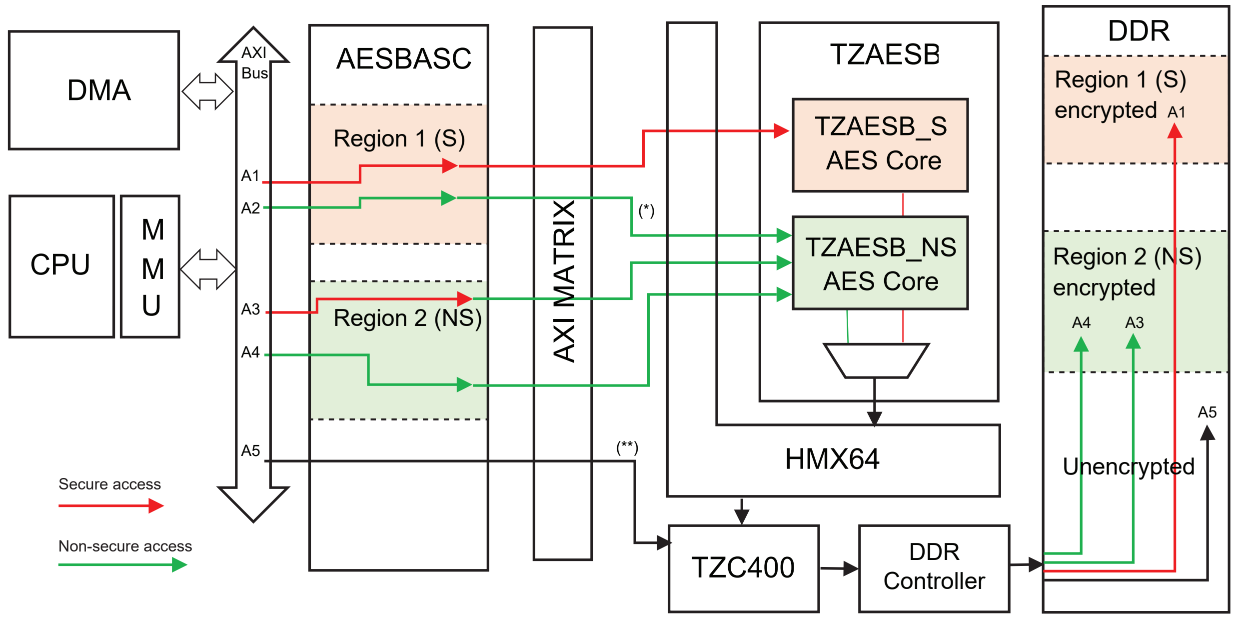
In order to use the TZAESB (for on-the-fly encrypted accesses), the TrustZone AES Bridge Address Space Controller (TZAESBASC) must be configured. This block defines the regions in the memory map where encryption is needed, and their security level. Whenever the address of a memory access matches a region defined in TZAESBASC, the access is routed through the TZAESB instead of taking the direct unencrypted path.

When access is performed to a region configured as non-secure in TZAESBASC, the protection bit is modified on-the-fly as non-secure even if the access was initially marked as secure. The access is then routed to the non-secure AES core in TZAESB (as illustrated with access A3 in the following figure).

TZAESBASC, MATRIX and TZC400 security settings must match.

The following schematic illustrates the behavior of the TZAESB when ID_TZAESB_S is secure and ID_TZAESB_NS is non-secure. This example shows five accesses (A1, A2, A3, A4, A5) coming either from the processor or the DMA with different securities, and targeting region 1 (secure), region 2 (non-secure) and a clear region (i.e., not defined in TZAESB) in the DDR space.

Figure 8-4. Example of Security Management with Different Access Types



Non-secure access A2 to region 1 (secure) is not immediately denied. The access is stopped when reaching TZC400, leading to a bus error with corresponding denied access information, due to TZC400 and TZAESBASC identical security settings.

Secure access A3 becomes non-secure when crossing TZAESBASC, leading TZAESB to select the non-secure AES core in order to use the correct encryption key. This access is not denied by TZC400 because the target address is a non-secure region.

When the application does not require two different encryption keys, the encrypted access bandwidth can be increased by running the two cores in parallel when two different hosts from different matrixes access TZAESB, for example DMAC0 and DMAC2. To do so, the user must program the same security bit values (ID_TZAESB_S or ID_TZAESB_NS) and use identical encryption keys for both cores.

8.4.7.2 Encryption Behavior

ID_TZAESB_S Security Bit Setting in TZPM	ID_TZAESB_NS Security Bit Setting in TZPM	Non-Secure Access to TZAESB_S User Interface	Non-Secure Access to TZAESB_NS User Interface	Non-Secure Memory Access through TZAESB	Secure Memory Access through TZAESB	Encryption Behavior
Secure	Non-secure	Denied	Accepted	Accepted; taking non-secure path; accesses to secure regions will be denied by TZC400	Accepted; only the accesses to secure regions take secure path	Secure regions encrypted with TZAESB_S key; non-secure regions encrypted with TZAESB_NS key Secure world can decrypt regions encrypted by Non-secure world even if keys differ.
Secure	Secure	Denied	Denied	Denied (response error)	Accepted only when targeting secure region[1]	Secure regions encrypted with improved bandwidth (TZAESB_S and TZAESB_NS keys must match)

...........continued

ID_TZAESB_S Security Bit Setting in TZPM	ID_TZAESB_NS Security Bit Setting in TZPM	Non-Secure Access to TZAESB_S User Interface	Non-Secure Access to TZAESB_NS User Interface	Non-Secure Memory Access through TZAESB	Secure Memory Access through TZAESB	Encryption Behavior
Non-secure	Non-secure	Accepted	Accepted	Accepted	Accepted	All regions encrypted with improved bandwidth (TZAESB_S and TZAESB_NS keys must match); Secure world can decrypt regions encrypted by Non-secure world as keys match.
Non-secure	Secure	NA	NA	NA	NA	Forbidden mode (not programmable)

Note:
1. Because the TZAESBASC converts secure accesses to non-secure accesses when the target address is in a non-secure region, the TZAESB denies the access if both security bits are secure.

8.4.7.3 Access Denials

ID_TZAESB_S Security Bit Setting in TZPM	ID_TZAESB_NS Security Bit Setting in TZPM	R/W Access to Secure Region from Non-Secure World	Bufferable Access (using MMU)	Data Abort	MATRIX Interrupt	TZC400 Interrupt
Secure	Non-secure	Read	–	Yes / No	Yes	Yes
		Write	No	Yes / No	Yes	Yes
			Yes	Yes / No	Yes(2) / No	Yes
Secure	Secure	Read	–	Yes	Yes	No(1)
		Write	No	Yes	Yes	No(1)
			Yes	Yes	Yes(2)	No(1)
Non-secure	Non-secure	Read	–	Yes / No	Yes	Yes
		Write	No	Yes / No	Yes	Yes
			Yes	Yes / No	Yes(2) / No	Yes

Notes:
1. Access denied by TZAESB, not propagated to TZC400.
2. In the case of write burst accesses, MATRIX_MEARx registers might not return the address corresponding to the offending access: due to the AXI protocol, the TZC400 response error indicates the last accessed address (not which of the consecutive accesses failed during the burst).

Peripheral ID	Type	Security
ID_AESBASC	User interface	Secure
ID_TZAESB_NS	User interface	Programmable Secure
ID_TZAESB_NS_INT	Interrupt ID only	Secure
ID_TZAESB_S	User interface	Programmable Secure
ID_TZAESB_S_INT	Interrupt ID only	Secure

Refer to the section TrustZone AESB Address Space Controller (TZAESBASC) for more details.

8.4.7.4 NAND Flash

NAND Flash pages can be accessed with TZAESB. ECC sections are not encrypted. With page sizes ≤ 4 Kbytes, use only bank 0 in NFC_RAM.

8.4.8 TrustZone Watchdog

8.4.8.1 Function

The lower 4 Kbytes of the user interface memory map are never secure and are dedicated to the Non-Secure Watchdog (NS_WDT). Above this offset, the user interface is dedicated to the Always Secure Watchdog (PS_WDT).

The TrustZone Watchdog offers a dedicated watchdog to both secure and non-secure worlds.

The TrustZone Watchdog offers a security escalation process based on transmitting interrupts before generating alarms or system reset.

Peripheral ID	Type	Security
ID_DWDT_S	User interface Interrupt ID	Secure (except the first 4 Kbytes)
ID_DWDT_NS	Interrupt ID only	Non-secure
ID_DWDT_ALARM	Interrupt ID only	Secure

Refer to the section Dual Watchdog Timer (DWDT) for more details.

8.4.8.2 Basic Programming Guidelines

The Configure Watchdog period is the time for a watchdog to generate an alarm or a system reset from the last counter reload. Note that watchdogs count on slow_clock/128, so the period step is about 4 ms.

The Configure Watchdog Interrupt level is the time for a watchdog to generate a “force” interrupt from the last counter reload. This time is necessarily shorter than the watchdog period. The interrupt level step is the same as the period step.

There are two PIT counters, one secure and one non-secure. These counters must be configured in a Continuous Interrupt mode, with a period shorter than the watchdog interrupt level. Note that PITs count on the system clock, so the step depends on the selected clock scheme.

PS_WDT must be configured to generate a reset or not when the counter underflows.

Security module TZWDT protection must be configured to erase or not the secure RAM in case of PS_WDT counter underflow.

PS_WDT and NS_WDT level interrupts (“force” interrupts) must be configured.

After a watchdog system reset, RSTC_SR can be read to get the watchdog reset flag. This register must be read before the watchdog reset event, since it only reports the reset event following the last read.

8.4.9 Security Module (SECUMOD)

SECUMOD_GPSBR is dedicated to control SECUMOD-related analog objects used for alarms or backup supply safety:

- Enable/disable backup power switch (PWSBU) switching to VBAT on a SECUMOD autobackup event
- Selection of VDDCPU supply monitor (SMCPU) range (depending on the processor frequency mode)
- Selection of temperature monitor (TEMPMON) high alarm threshold (105°C/120°C typ)
- Selection of external interrupts EXT_IRQ0/EXT_IRQ1 for Automatic Backup mode protocol

Two independent peripheral IDs are available for either SECUMOD or SECURAM.

Peripheral ID	Type	Security
ID_SECUMOD	User interface interrupt	Always Secure
ID_SECURAM	Interrupt only	Always Secure

Refer to the section Security Module (SECUMOD) for more details.

8.4.10 Extended DMA Controller (XDMAC)

Security of DMA channels is configurable channel-by-channel using bit PROT in the XDMAC_CC register. This bit can be configured in the Secure world only. A channel is non-secure by default.

Interrupts related to secure DMA channels are routed on a secure interrupt line.

Peripheral ID	Type	Security
ID_XDMAC0	User interface interrupt	Programmable Secure
ID_XDMAC0_SINT	Interrupt only (Secure channel)	Secure
ID_XDMAC1	User interface interrupt	Programmable Secure
ID_XDMAC1_SINT	Interrupt only (Secure channel)	Secure
ID_XDMAC2	User interface interrupt	Programmable Secure
ID_XDMAC2_SINT	Interrupt only (Secure channel)	Secure

Refer to the section DMA Controller (XDMAC) for more details.

8.5 Peripheral Clocks and Security

Table 8-11. Peripheral Identifiers

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
0	DWDT	AS	–	SW	–	–	MCK0	–	–	–	–	–	–	–	Dual Watchdog Timer, Secure World
1	DWDT	NS	–	NSW	–	–	MCK0	–	–	–	–	–	–	–	Dual Watchdog Timer, Non-secure World
2	DWDT	AS	–	NSW_ALARM	–	–	MCK0	–	–	–	–	–	–	–	Dual Watchdog Timer, Non-secure World Alarm
4	SCKC	AS	–	–	–	–	MCK0	–	–	–	–	–	–	–	Slow Clock Controller
5	SHDWC	AS	–	–	–	–	MCK0	–	–	–	–	–	–	–	Shutdown Controller
6	RSTC	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Reset Controller
7	RTC	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Real-Time Clock
8	RTT	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Real-Time Timer
9	CHIPID	PS	TZPM	–	–	–	MCK0	–	–	–	–	–	–	–	Chip Identifier
10	PMC	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Power Management Controller
11	PIOA	PS	PIOA	X	–	X	MCK0	–	–	–	–	–	–	–	For PIO 0 to 31
12	PIOB	PS	PIOB	X	–	–	MCK0	–	–	–	–	–	–	–	For PIO 32 to 63
13	PIOC	PS	PIOC	X	–	–	MCK0	–	–	–	–	–	–	–	For PIO 64 to 95
14	PIOD	PS	PIOD	X	–	–	MCK0	–	–	–	–	–	–	–	For PIO 96 to 127
15	PIOE	PS	PIOE	X	–	–	MCK0	–	–	–	–	–	–	–	For PIO 128 to 135
17	SECUMOD	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Security Module
18	SECURAM	AS	–	X	–	–	MCK0	–	–	–	–	–	–	–	Secure Backup SRAM
19	SFR	PS	TZPM	–	–	X	MCK1	–	–	–	–	–	–	–	Special Function Register

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
20	SFRBU	AS	–	–	–	–	MCK0	–	–	–	–	–	–	–	Special Function Register in Backup zone
21	HSMC	PS	MATRIX + TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Static Memory Controller – NAND Flash Controller
22	XDMAC0	PS	XDMAC0	X	–	X	MCK1	–	–	–	–	–	–	–	DMA 0, mem to periph, 32 channels
23	XDMAC1	PS	XDMAC1	X	–	X	MCK1	–	–	–	–	–	–	–	DMA 1, mem to periph, 32 channels
24	XDMAC2	PS	XDMAC2	X	–	X	MCK1	–	–	–	–	–	–	–	DMA 2, mem to mem, 8 channels
25	ACC	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Analog Comparator Controller
26	ADC	PS	TZPM	X	–	–	GCLK(1)	100	X	–	–	X	X	–	Analog-to-Digital Converter
27	AES	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Advanced Encryption Standard
28	TZAESBASC	AS	TZPM	–	–	X	MCK1	–	–	–	–	–	–	–	TrustZone Advanced Encryption Standard Bridge – Address Space Controlller
29	ARM	PS	MMU	–	–	–	MCK1	50	X	X	–	–	X	–	Cortex-A7 Core 0
30	ASRC	PS	TZPM	X	–	X	MCK1	200	X	–	–	–	X	–	Asynchronous Sample Rate Converter
32	CPKCC	PS	TZPM	X	–	X	MCK0	–	–	–	–	–	–	–	Classic Public Key Cryptography Controller
33	CSI	PS	TZPM	X	–	X	MCK3	27	X	–	X	–	–	–	Camera Serial Interface 2 between ISC and MIPI PHY

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
34	CSI2DC	PS	TZPM	X	–	X	MCK3	–	–	–	–	–	–	–	CSI to Demultiplexer Controller
35	DDR3PHY	PS	TZC + TZPM	–	–	–	MCK2	–	–	–	–	–	–	–	DDR/LPDDR Physical Interface
36	UDDRC	PS	TZC + TZPM	–	–	–	MCK2	–	–	–	–	–	–	–	Universal DDR Memory Controller
37	EIC	PS	TZPM	–	–	X	MCK1	–	–	–	–	–	–	–	External Interrupt Controller
38	FLEXCOM0	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 0
39	FLEXCOM1	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 1
40	FLEXCOM2	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 2
41	FLEXCOM3	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 3
42	FLEXCOM4	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 4
43	FLEXCOM5	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 5
44	FLEXCOM6	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 6
45	FLEXCOM7	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 7
46	FLEXCOM8	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 8
47	FLEXCOM9	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 9
48	FLEXCOM10	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	–	X	–	–	FLEXCOM 10
49	FLEXCOM11	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–		X	–	–	FLEXCOM 11
51	GMAC0	PS	TZPM	X	–	X	MCK1	125	–	–	–	–	–	X	Gigabit Ethernet MAC
52	GMAC1	PS	TZPM	X	–	X	MCK1	50	–	–	–	–	–	X	Ethernet MAC
53	GMAC0	PS	same as GMAC0	TSU	–	–	MCK1	200	–	–	–	–	X	X	Gigabit Ethernet MAC – Timestamp Unit Generic Clock – No Interrupt
54	GMAC1	PS	same as GMAC1	TSU	–	–	MCK1	200	–	–	–	–	X	X	Ethernet MAC – Timestamp Unit Generic Clock – No Interrupt

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
55	ICM	AS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Integrity Check Monitor
56	ISC	PS	TZPM	X	–	X	MCK3	–	–	–	–	–	–	–	Camera Interface
57	I2SMCC0	PS	TZPM	X	–	X	MCK1	100	X	–	–	–	X	–	Inter-IC Sound Controller 0
58	I2SMCC1	PS	TZPM	X	–	X	MCK1	100	X	–	–	–	X	–	Inter-IC Sound Controller 1
60	MATRIX	AS	–	X	–	–	MCK1	–	–	–	–	–	–	–	AHB Matrix
61	MCAN0	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 0
62	MCAN1	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 1
63	MCAN2	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 2
64	MCAN3	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 3
65	MCAN4	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 4
66	MCAN5	PS	TZPM	INT0	–	X	MCK1	80	X	–	–	X	–	–	Host CAN 5
67	OTPC	PS	TZPM	X	–	–	MCK0	–	–	–	–	–	–	–	One Time Programmable Memory Controller
68	PDMC0	PS	TZPM	X	–	X	MCK1	50	X	–	–	–	X	–	Pulse Density Modulation Interface Controller 0
69	PDMC1	PS	TZPM	X	–	X	MCK1	50	X	–	–	–	X	–	Pulse Density Modulation Interface Controller 1
70	PIT64B0	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 0
71	PIT64B1	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 1
72	PIT64B2	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 2
73	PIT64B3	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 3

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
74	PIT64B4	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 4
75	PIT64B5	PS	TZPM	X	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	64-bit Periodic Interval Timer 5
77	PWM	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Pulse Width Modulation
78	QSPI0	PS	MATRIX + TZPM	X	–	X	MCK1	200	X	–	–	X	–	–	Quad IO Serial Peripheral Interface 0
79	QSPI1	PS	MATRIX + TZPM	X	–	X	MCK1	200	X	–	–	X	–	–	Quad IO Serial Peripheral Interface 1
80	SDMMC0	PS	TZPM	X	–	X	MCK1	208	X	–	X	X	–	X	Ultra High Speed SD Host Controller 0 (e.MMC 5.1)
81	SDMMC1	PS	TZPM	X	–	X	MCK1	208	X	–	X	X	–	X	Ultra High Speed SD Host Controller 1 (e.MMC 4.51)
82	SDMMC2	PS	TZPM	X	–	X	MCK1	208	X	–	X	X	–	X	Ultra High Speed SD Host Controller 2 (e.MMC 4.51)
83	SHA	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Secure Hash Algorithm
84	SPDIFRX	PS	TZPM	X	–	X	MCK1	150	X	–	–	–	X	–	Sony Philips Digital Interface RX
85	SPDIFTX	PS	TZPM	X	–	X	MCK1	25	X	–	–	–	X	–	Sony Philips Digital Interface TX
86	SSC0	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Synchronous Serial Interface 0
87	SSC1	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Synchronous Serial Interface 1
88	TC0	PS	TZPM	CHANNEL0	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	32-bit Timer Counter 0 Channel 0

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
89	TC0	PS	same as channel0	CHANNEL1	–	X	MCK1	–	–	–	–	–	–	–	32-bit Timer Counter 0 Channel 1
90	TC0	PS	same as channel0	CHANNEL2	–	X	MCK1	–	–	–	–	–	–	–	32-bit Timer Counter 0 Channel 2
91	TC1	PS	TZPM	CHANNEL0	–	X	MCK1	$f_{MCK1}/3$	X	–	X	X	X	X	32-bit Timer Counter 1 Channel 0
92	TC1	PS	same as channel0	CHANNEL1	–	X	MCK1	–	–	–	–	–	–	–	32-bit Timer Counter 1 Channel 1
93	TC1	PS	same as channel0	CHANNEL2	–	X	MCK1	–	–	–	–	–	–	–	32-bit Timer Counter 1 Channel 2
94	TCPCA	PS	TZPM	X	–	X	MCK1	Note 1	–	–	–	–	–	–	USB Type-C Port Controller A
95	TCPCB	PS	TZPM	X	–	X	MCK1	Note 1	–	–	–	–	–	–	USB Type-C Port Controller B
96	TDES	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	Triple Data Encryption Standard
97	TRNG	PS	TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	True Random Number Generator
98	TZAESB	PS	TZPM	NS	–	X	MCK1	–	–	–	–	–	–	–	TrustZone Advanced Encryption Standard Bridge Non–Secure (Clocks & Interrupt)
99	TZAESB	AS	–	NS_INT	–	–	MCK1	–	–	–	–	–	–	–	TrustZone Advanced Encryption Standard Bridge Non–Secure (Interrupt only)

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
100	TZAESB	PS	TZPM	S	–	–	MCK1	–	–	–	–	–	–	–	TrustZone Advanced Encryption Standard Bridge Secure (Interrupt only)
101	TZAESB	AS	–	S_INT	–	–	MCK1	–	–	–	–	–	–	–	TrustZone Advanced Encryption Standard Bridge Secure (Interrupt only)
102	TZC	AS	AS	X	–	–	MCK1	–	–	–	–	–	–	–	TrustZone Address Space Controller (TZC400)
104	UDPHSA	PS	MATRIX + TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	USB Device High Speed A
105	UDPHSB	PS	MATRIX + TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	USB Device High Speed B
106	UHPHS	PS	MATRIX + TZPM	X	–	X	MCK1	–	–	–	–	–	–	–	USB Host Controller High Speed
110	ARM	PS	MMU	nPMUIRQ	–	–	MCK1	–	–	–	–	–	–	–	Performance Monitoring Unit
111	ARM	PS	MMU	nAXIERRIRQ	–	–	MCK1	–	–	–	–	–	–	–	AXI Transaction Error
112	XDMAC0	PS	XDMAC0	SINT	–	–	MCK1	–	–	–	–	–	–	–	DMA0, mem to periph, 32 channels, Secure Interrupt
113	XDMAC1	PS	XDAMC1	SINT	–	–	MCK1	–	–	–	–	–	–	–	DMA1, mem to periph, 32 channels, Secure Interrupt
114	XDMAC2	PS	XDMAC2	SINT	–	–	MCK1	–	–	–	–	–	–	–	DMA2, mem to mem, 8 channels, Secure Interrupt

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
115	AES	PS	same as AES	SINT	–	–	MCK1	–	–	–	–	–	–	–	Advanced Encryption Standard, Secure Interrupt
116	GMAC0	PS	same as GMAC0	Q1	–	–	MCK1	–	–	–	–	–	–	–	GMAC0 Queue 1 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 1
117	GMAC0	PS	same as GMAC0	Q2	–	–	MCK1	–	–	–	–	–	–	–	GMAC0 Queue 2 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 2
118	GMAC0	PS	same as GMAC0	Q3	–	–	MCK1	–	–	–	–	–	–	–	GMAC0 Queue 3 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 3
119	GMAC0	PS	same as GMAC0	Q4	–	–	MCK1	–	–	–	–	–	–	–	GMAC0 Queue 4 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 4
120	GMAC0	PS	same as GMAC0	Q5	–	–	MCK1	–	–	–	–	–	–	–	GMAC0 Queue 5 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 5

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
121	GMAC1	PS	same as GMAC1	Q1	–	–	MCK1	–	–	–	–	–	–	–	GMAC1 Queue 1 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 1
122	ICM	AS	same as ICM	SINT	–	–	MCK1	–	–	–	–	–	–	–	Integrity Check Monitor, Secure Interrupt
123	MCAN0	PS	same as MCAN0	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN0 Interrupt1
124	MCAN1	PS	same as MCAN1	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN1 Interrupt1
125	MCAN2	PS	same as MCAN2	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN2 Interrupt1
126	MCAN3	PS	same as MCAN3	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN3 Interrupt1
127	MCAN4	PS	same as MCAN4	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN4 Interrupt1
128	MCAN5	PS	same as MCAN5	INT1	–	–	MCK1	–	–	–	–	–	–	–	MCAN5 Interrupt1
129	PIOA	PS	same as PIOA	SINT	–	–	MCK0	–	–	–	–	–	–	–	For PIO 0 to 31, Secure Interrupt
130	PIOB	PS	same as PIOB	SINT	–	–	MCK0	–	–	–	–	–	–	–	For PIO 32 to 63, Secure Interrupt
131	PIOC	PS	same as PIOC	SINT	–	–	MCK0	–	–	–	–	–	–	–	For PIO 64 to 95, Secure Interrupt
132	PIOD	PS	same as PIOD	SINT	–	–	MCK0	–	–	–	–	–	–	–	For PIO 96 to 127, Secure Interrupt
133	PIOE	PS	same as PIOE	SINT	–	–	MCK0	–	–	–	–	–	–	–	For PIO 128 to 136, Secure Interrupt
140	PIT64B5	PS	same as PIT64B5	SINT	–	–	MCK1	–	–	–	–	–	–	–	64-bit Periodic Interval Timer 5, Secure Interrupt

...........continued

Instance ID	Instance Name	Security	TZ Security Management	GIC SPI Interrupt	External Interrupt	PMC Clock Control	Main System Bus Clock	Max Generic Clock Freq. (MHz)	SYSPLLCK	DDRPLLCK	IMGPLLCK	BAUDPLLCK	AUDIOPLLCK	ETHPLLCK	Instance Description
141	SDMMC0	PS	same as SDMMC0	TIMER	–	–	MCK1	–	–	–	–	–	–	–	Ultra High Speed SD Host Controller 0 (e.MMC 5.1) Timer interrupt
142	SDMMC1	PS	same as SDMMC1	TIMER	–	–	MCK1	–	–	–	–	–	–	–	Ultra High Speed SD Host Controller 1 (e.MMC 4.51) Timer interrupt
143	SDMMC2	PS	same as SDMMC2	TIMER	–	–	MCK1	–	–	–	–	–	–	–	Ultra High Speed SD Host controller 2 (e.MMC 4.51) Timer interrupt
144	SHA	PS	same as SHA	SINT	–	–	MCK1	–	–	–	–	–	–	–	Secure Hash Algorithm, Secure Interrupt
151	TDES	PS	same as TDES	SINT	–	–	MCK1	–	–	–	–	–	–	–	Triple Data Encryption Standard, Secure Interrupt
152	TRNG	PS	same as TRNG	SINT	–	–	MCK1	–	–	–	–	–	–	–	True Random Number Generator, Secure Interrupt
153	EIC	PS	same as EIC	X	EXT_IRQ0	–	MCK1	–	–	–	–	–	–	–	External Interrupt ID0
154	EIC	PS	same as EIC	X	EXT_IRQ1	–	MCK1	–	–	–	–	–	–	–	External Interrupt ID1

⚠WARNING

1. GCLK must be started before accessing registers.

Note:

1. Select GCLK input as MD_SLCK or TD_SLCK.

9. Cortex-A7 Processor (Arm)

9.1 Description

The high-performance, low-power Arm Cortex-A7 processor features an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A7 processor implements the Arm v7-A architecture and runs 32-bit Arm instructions, and 16-bit and 32-bit Thumb instructions.

The Cortex-A7 NEON Media Processing Engine (MPE) extends the Cortex-A7 functionality to provide support for the Arm v7 Advanced SIMD v2 and Vector Floating-Point v4 (VFPv4) instruction sets. The Cortex-A7 NEON MPE provides flexible and powerful acceleration for signal processing algorithms including multimedia such as image processing, video decode/encode, 2D/3D graphics, and audio. For details, refer to the Cortex-A7 NEON Media Processing Engine Technical Reference Manual.

The Cortex-A7 processor includes TrustZone technology to enhance security by partitioning the SoC's hardware and software resources in a Secure world for the security subsystem and a Normal world for the rest, enabling a strong security perimeter to be built between the two. For details, refer to Security Extensions Overview in the Cortex-A7 Technical Reference Manual. Refer to the Arm Architecture Reference Manual, Arm v7-A and Arm v7-R edition for details on how TrustZone works in the architecture.

The Cortex-A7 core, the Floating Point Unit (FPU) and NEON Media Processing Engine (MPE) are based on Arm Ltd. r0p5 and are highly configurable at the hardware design stage. These configurations are described in the sections that follow. The default software configuration has been intensively tested and leads to best results in any conditions.

9.1.1 Reference Documents

Document Title	Available
Cortex-A7 Core Technical Reference Manual	developer.arm.com/
Cortex-A7 Floating-Point Unit Technical Reference Manual	
Cortex-A7 NEON Media Processing Engine Technical Reference Manual	
Arm Architecture Reference Manual Arm v7-A and Arm v7-R edition	
Coresight 400 Technical Reference Manual	

9.2 Embedded Characteristics

The Cortex-A7 processor (r0p5) implements the Arm v7-A architecture. This includes:

- 32-bit Arm Instruction Set
- Thumb Instruction Set featuring 16-bit and 32-bit Instructions
- ThumbEE Instruction Set
- Implementation of the Jazelle Extension
- Arm v7 Debug Architecture
- TrustZone Security Extensions
- Harvard Level 1 Memory System with a Memory Management Unit (MMU)
- 32 Kbytes L1 Data Cache
- 32 Kbytes L1 Instruction Cache
- 256 Kbytes L2 Cache
- Generic Interrupt Controller (GIC)
- Media Processing Engine (MPE) with NEON Technology
- Trace Support through an Embedded Trace Macrocell (ETM) Interface

- Performance Monitoring Unit (PMU)

9.3 Clocks

CPUPLLCK, comprising the L2 cache controller and L2 cache memory, feeds the Cortex-A7 processor, the L1 cache, the NEON MPE, the ETM and the GIC.

CPUPLLCK is divided (MCK0) to feed the rest of the core subsystem and the CSS AXI matrix.

The L2 cache controller runs at CPU frequency. The L2 cache memories run at one half of the CPU clock frequency.

The CPU frequency can be adapted to the application requirements with frequency and voltage scaling, including power saving, in case high performance is not required. Refer to VDDCPU characteristics in the section Electrical Characteristics for more details.

To optimize performance, a path dedicated to high frequency exists between the CPU and UDDRC. It is clocked by MCK4. Refer to the section System Interconnect and Security (SIS) for more details.

9.4 Power Supplies

The Cortex-A7 processor, including L1 and L2 caches, is powered with dedicated VDDCPU power rails, while the rest of the core system is powered with VDDCORE.

By separating the power supplies, power consumption can be reduced when the core is not active by powering off the VDDCPU section. Refer to ULP2 Low Power Mode in the section Electrical Characteristics for more details.

The CPU power supply can be adapted to the requirements with power scaling in case high performance is required. Refer to VDDCPU characteristics in the section Electrical Characteristics for more details.

9.5 Generic Interrupt Controller (GIC)

The integrated Generic Interrupt Controller (GIC) collates and arbitrates from a large number of interrupt sources. It provides:

- Masking of interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target processors
- Tracking the status of interrupts
- Generation of interrupts by software
- Support for security extensions
- Support for virtualization extensions

Refer to the Cortex-A7 Core Technical Reference Manual for more information about the GIC.

9.6 Media Processing Engine (MPE) with NEON Technology

The Cortex-A7 NEON Media Processing Engine (MPE) extends the Cortex-A7 functionality to provide support for the Arm v7 Advanced SIMDv2 and Vector Floating-Pointv4 (VFPv4) instruction sets. The Cortex-A7 NEON MPE supports all addressing modes and data-processing operations described in the Arm Architecture Reference Manual, Arm v7-A and Arm v7-R edition.

The Cortex-A7 NEON MPE includes the following features:

- SIMD and scalar single-precision floating-point computation
- Scalar double-precision floating-point computation
- SIMD and scalar half-precision floating-point conversion
- SIMD 8, 16, 32, and 64-bit signed and unsigned integer computation
- 8 or 16-bit polynomial computation for single-bit coefficients

- Structured data load capabilities
- Large, shared register file, addressable as:
  - Thirty-two 32-bit S (single) registers
  - Thirty-two 64-bit D (double) registers
  - Sixteen 128-bit Q (quad) registers

See the Arm Architecture Reference Manual, Arm v7-A and Arm v7-R edition for more information about the extension register set.

The operations include:

- Addition and subtraction
- Multiplication with optional accumulation
- Maximum or minimum value driven lane selection operations
- Inverse square-root approximation
- Comprehensive data-structure load instructions, including register-bank-resident tablelookup

See the Cortex-A7 NEON Media Processing Engine Technical Reference Manual for details of the NEON extensions.

9.7 Debug

Refer to the section Debug and Test for full details.

9.8 Performance Monitoring Unit (PMU)

The processor features a Performance Monitoring Unit (PMU), made up of logic to gather various statistics on the operation of the processor and memory system during runtime, based on the PMUv2 architecture. These events provide useful information about the behavior of the processor that can be used when debugging or profiling code.

The PMU provides four counters. Each counter can count any of the events available in the processor.

Refer to the Cortex-A7 Core Technical Reference Manual for details on the PMU extensions.

9.9 Timestamp Generator

The Arm ID (GCLK29) is the source clock for the timestamp generator. It is running and is configured to use MAINCK with no divider at reset.

The Coresight 400 IP is memory-mapped into APB_DEBUG_S (0xE8800000) and the timestamp registers are at an offset of 0x43000.

To generate timestamps:

1. , access the memory map interface of the coresight timestamp module[(1)] and:
   a. Set the timestamp generator frequency value to 24 MHz by writing 24000000 to the CNTFID0 register (same as the MAINCK frequency).
   b. Enable the timestamp generator by writing 1 to the CNTCR register.
2. Enable PPI interrupt ID 29 in GIC (secure physical timer interrupt)[(2)].
3. Access the generic timer coprocessor interface (CP15)[(3)] to:
   a. Set the counter frequency value to 24 MHz in the CNTFRQ register.
   b. Set the timer count value in the CNTP_TVAL register.
   c. Enable the Timer event by writing into the CNTP_CTL register.

When the timer expires, a PPI interrupt is generated with interrupt ID 29.

Notes:

1. Refer to Arm V7A, ARCH Manual (DDI0406C), Appendix D5.2, Coresight 400 TRM (DDI0480G), Section 3.19.
2. Refer to Cortex-A7 TRM (DDI0464F), Section 8.2.2.
3. Refer to Arm V7A, ARCH Manual (DDI0406C), B8.2.

10. External Interrupt Controller (EIC)

10.1 Description

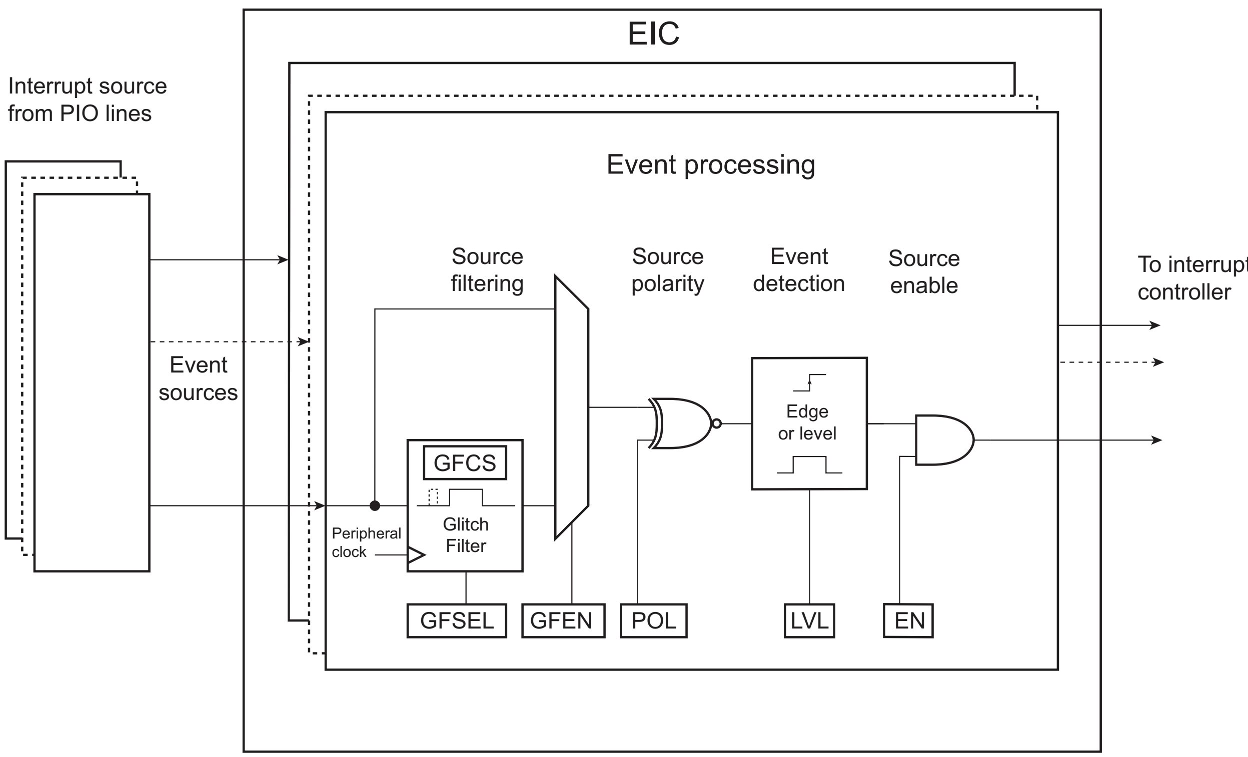
The External Interrupt Controller (EIC) forwards external interrupt sources from two PIO lines to the interrupt controller by providing some pre-processing capabilities (filtering, polarity inversion, edge detection, enable).

10.2 Embedded Characteristics

- Glitch Filter
- Programmable Source Polarity
- Edge or Level Detection
- Freeze Capability for Each Interrupt Line Configuration
- Global Configuration Write Protection Register with Key

10.3 Block Diagram

Figure 10-1. EIC Block Diagram



10.4 I/O Lines Description

Pin Name	Pin Description	Type
EIC_EIS0–EIC_EIS1	External Interrupt Source 0–External Interrupt Source 1	Input

10.5 Product Dependencies

10.5.1 I/O Lines

The external interrupt signals EIC_EIS0..EIC_EIS1 are multiplexed through the PIO controller(s).

Depending on the features of the PIO controller used in the product, the pins must be programmed in accordance with their assigned interrupt function.

10.5.2 Power Management

The EIC is continuously clocked. The Power Management Controller has no effect on the EIC behavior.

10.5.3 External Interrupt Sources

The following table describes the external interrupt sources available in the product.

Source Number	Source Name	Source Description
0	EIC_EIS0	External Interrupt Source
1	EIC_EIS1	External Interrupt Source

10.6 Functional Description

10.6.1 Interrupt Line Characteristics

For each event source, the EIC features a dedicated interrupt line as shown in the Block Diagram.

If Glitch Filter Enable (GFEN) is cleared in the Source Configuration register (EIC_SCFGxR), pulses of any width can be forwarded and detected as valid source events on the interrupt line x regardless of the respective frequencies of the source clock, the EIC peripheral clock and the Interrupt Controller clock.

If EIC_SCFGxR.GFEN is set, the source events are filtered to remove unwanted glitches. The glitch filter forwards its input level when it is maintained for more than 2^{GFSEL} peripheral clock cycles and rejects pulses narrower than 2^{GFSEL}-1 peripheral clock cycles.

The source event polarity can be changed by EIC_SCFGxR.POL.

Either edge or level event detection can be performed by EIC_SCFGxR.LVL.

When set, EIC_SCFGxR.FRZ freezes all interrupt line settings until hardware reset. This includes the settings in EIC_SCFGxR. This feature can be used to prevent any corruption of the critical interrupt lines configuration.

If the glitch filter is enabled in EIC_SCFGxR.GFEN, the maximum propagation delay through the EIC (see the following figure) is calculated as:

$$\mathrm{EIC\_delay\_max} = (2 + 2\mathrm{GFSEL}) * \mathrm{periph\_clk\_period} + 2.5 * \mathrm{clock\_period}$$

and the minimum propagation delay through the EIC is:

$$\mathrm{EIC\_delay\_min} = (1 + 2\mathrm{GFSEL}) * \mathrm{periph\_clk\_period} + 1 * \mathrm{clock\_period}$$

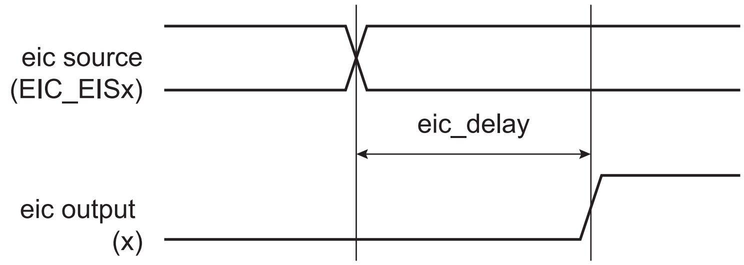
If the glitch filter is disabled, the maximum propagation delay through the EIC is calculated as:

$$\mathrm{EIC\_delay\_max} = 2.5 * \mathrm{clock\_period}$$

and the minimum propagation delay through the EIC is:

$$\mathrm{EIC\_delay\_min} = 1 * \mathrm{clock\_period}$$

Figure 10-2. EIC Propagation Delay



clock_period is the peripheral clock period of the EIC as programmed in the Power Management Controller.

10.6.2 External Interrupt Line Programming

The steps to set up or change the external interrupt line x configuration are the following:

1. Clear SCFGxR.EN to disable the EIC interrupt source.
2. Perform any external setup that might be required at the source peripheral and wait until the peripheral source output is stabilized.
3. If the glitch filter is enabled on the interrupt line, wait until the Ready flag RDYx is set in the Glitch Filter Configuration Status register (EIC_GFCS).
4. Write the required interrupt line configuration fields among GFSEL, GFEN, POL, LVL into EIC_SCFGxR.
5. If the glitch filter is enabled on the interrupt line, wait until EIC_GFCS.RDYx is set.
6. Set EIC_SCFGxR.EN if the source is not yet enabled and, if required for safety, freeze the interrupt line x configuration by setting EIC_SCFGxR.FRZ.

Example:

If the user desires to stop or minimize the circuit activity after a voltage or frequency failure detection on a rising edge of an EIC interrupt source, the polarity can be changed with a single write to EIC_SCFGxR.POL to detect a falling edge of the same source. Detection of the falling edge source indicates the recovery of a normal condition. In this case, the steps described previously can be ignored.

10.6.3 Register Write Protection

To prevent any single software error from corrupting EIC behavior, certain registers in the address space can be write-protected.

The following register can be write-protected by setting WPCFEN in the Write Protection Mode register (EIC_WPMR):

- External Interrupt Controller Source Configuration register x (EIC_SCFGxR)

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status register (EIC_WPSR) is set and WVSRC indicates the register in which the write access was attempted.

WPVS is automatically cleared after reading EIC_WPSR.

If a write attempts to modify the settings of an interrupt line x that was previously frozen by setting EIC_SCFGxR.FRZ, the FZWVS flag in EIC_WPSR is set and WVSRC indicates the register in which the write access was attempted.

FZWVS is automatically cleared after reading EIC_WPSR.

If a write attempts to modify the settings of the glitch filter of interrupt line x but the Source Configuration register x (EIC_SCFGxR) access is discarded due to an already ongoing glitch filter configuration, the FZWVS flag in EIC_WPSR is set and WVSRC indicates the register in which the write access was attempted.

FZWVS is automatically cleared after reading EIC_WPSR.

10.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	EIC_GFCS	31:24								
		23:16								
		15:8								
		7:0							RDY1	RDY0
0x04	EIC_SCFG0R	31:24	FRZ							
		23:16								EN
		15:8							LVL	POL
		7:0				GFEN			GFSEL[1:0]	
0x08	EIC_SCFG1R	31:24	FRZ							
		23:16								EN
		15:8							LVL	POL
		7:0				GFEN			GFSEL[1:0]	
0x0C ... 0xE3	Reserved									
0xE4	EIC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPCFEN
0xE8	EIC_WPSR	31:24								
		23:16								
		15:8	WVSRC[7:0]							
		7:0						BSWVS	FZWVS	WPVS

10.7.1 EIC Glitch Filter Configuration Status Register

Name: EIC_GFCS
Offset: 0x00
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RDY1	RDY0
Access							R	R
Reset							0	0

Bits 0, 1 – RDYx Filter x Configuration Ready

Value	Description
0	The interrupt line x glitch filter is not yet ready for use due to a previous write to EIC_SCFGxR, or the glitch filter is not implemented for this interrupt line. The glitch filter must not be reprogrammed in EIC_SCFGxR.
1	The interrupt line x glitch filter configuration is done and the glitch filter is active. The glitch filter can be programmed in EIC_SCFGxR.

10.7.2 EIC Source Configuration Register x

Name: EIC_SCFGxR
Offset: 0x04 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFEN is cleared in the EIC Write Protection Mode register.

Bit	31	30	29	28	27	26	25	24
	FRZ							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
								EN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							LVL	POL
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
				GFEN			GFSEL[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 31 – FRZ Interrupt Line Freeze

Value	Description
0	No effect.
1	EIC_SCFGxR is frozen until hardware reset.

Bit 16 – EN Source Enable

Value	Description
0	The EIC source x is disabled. Any source edge or level detection is discarded.
1	The EIC source x is enabled.

Bit 9 – LVL Level Detection

Value	Description
0	The EIC source x interrupt status is set on a valid source edge.
1	The EIC source x interrupt status is set on a valid source level.

Bit 8 – POL Polarity

Value	Description
0	The EIC source x is active low if LVL is set, or active on falling edge if LVL is cleared.
1	The EIC source x is active high if LVL is set, or active on rising edge if LVL is cleared.

Bit 4 – GFEN Glitch Filter Enable

Value	Description
0	The glitch filter is disabled or not implemented for EIC source x. Any source x change is forwarded as is to the source detection logic.
1	The glitch filter is enabled for EIC source x. The EIC source x glitches are filtered according to Interrupt Line Characteristics.

Bits 1:0 – GFSEL[1:0] Glitch Filter Selector
If GFEN is set, EIC source x glitches are filtered according to Interrupt Line Characteristics.

10.7.3 EIC Write Protection Mode Register

Name: EIC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								WPCFEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x454943	PASSWD	Writing any other value in this field aborts the write operation of the WPCFEN bit. Always reads as 0.

Bit 0 – WPCFEN Write Protection Configuration Enable

Value	Description
0	Disables the write protection on the Configuration registers if WPKEY corresponds to 0x454943 (“EIC” in ASCII).
1	Enables the write protection on the Configuration registers if WPKEY corresponds to 0x454943 (“EIC” in ASCII).

10.7.4 EIC Write Protection Status Register

Name: EIC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						BSWVS	FZWVS	WPVS
Access						R	R	R
Reset						0	0	0

Bits 15:8 – WVSRC[7:0] Write Violation Source
When either WPVS, FZWVS or BSWVS is set to 1, WVSRC indicates the register address offset at which a write access has been attempted.

Bit 2 – BSWVS Busy Register Write Violation Status

Value	Description
0	No write access violation of a busy register has occurred since the last read of EIC_WPSR.
1	A write access violation of a busy register has occurred since the last read of EIC_WPSR. The associated violation is reported into WVSRC.

Bit 1 – FZWVS Frozen Register Write Violation Status

Value	Description
0	No write access violation of a frozen register has occurred since the last read of EIC_WPSR.
1	A write access violation of a frozen register has occurred since the last read of EIC_WPSR. The associated violation is reported into WVSRC.

Bit 0 – WPVS Write Protection Register Violation Status

Value	Description
0	No write protection violation has occurred since the last read of EIC_WPSR.
1	A write protection violation has occurred since the last read of EIC_WPSR. The associated violation is reported into WVSRC.

11. Debug and Test

11.1 Description

The product features a number of complementary debug and test capabilities.

A common JTAG/ICE (In-Circuit Emulator) port is used for standard debugging functions, such as downloading code and single-stepping through programs.

A 2-pin debug port Serial Wire Debug (SWD) is provided in addition to the 5-pin JTAG port. The SWD provides an easy and risk-free alternative to JTAG as the two signals, SWDIO and SWCLK, are overlaid on the TMS and TCK pins, allowing for bi-modal devices that provide the other JTAG signals. These extra JTAG pins can be switched to other uses when in SWD mode.

A set of debug and test input/output pins gives direct access to these capabilities from a PC-based test environment.

11.1.1 Reference Documents

The Cortex-A7 core is based on Arm Ltd. r0p5 and is highly configurable at the hardware design stage. These configurations are described in the sections that follow.

Document Title	Available
Cortex-A7 Core Technical Reference Manual	developer.arm.com/
Arm Architecture Reference Manual Arm v7-A and Arm v7-R edition	

11.2 Embedded Characteristics

- Cortex-A7 Processor In the Loop (PIL)
  - Cortex-A7 uniprocessor
  - ETM-A7 (Trace Macrocell)
  - CTI (Cross-Triggering Interface) module
  - CTM (Cross-Triggering Matrix) module
  - APB subsystem
  - ROM table (processor ROM table)
- CSSYS Block (Debug and Trace Subsystem from SoC-400)
  - DAP with an asynchronous bridge to allow DAP/CPU asynchronous clock
  - CXAPBIC APB interconnect containing the top-level ROM table
  - ETB (Embedded Trace Buffer + 16-Kbyte ETB RAM)
  - Timestamp generation component
  - ECT (Embedded Cross-Triggering) component
- Chip ID Register
- IEEE 1149.1 JTAG Boundary-scan on All Digital Pins

11.3 Block Diagram

Figure 11-1. Debug and Test General Block Diagram

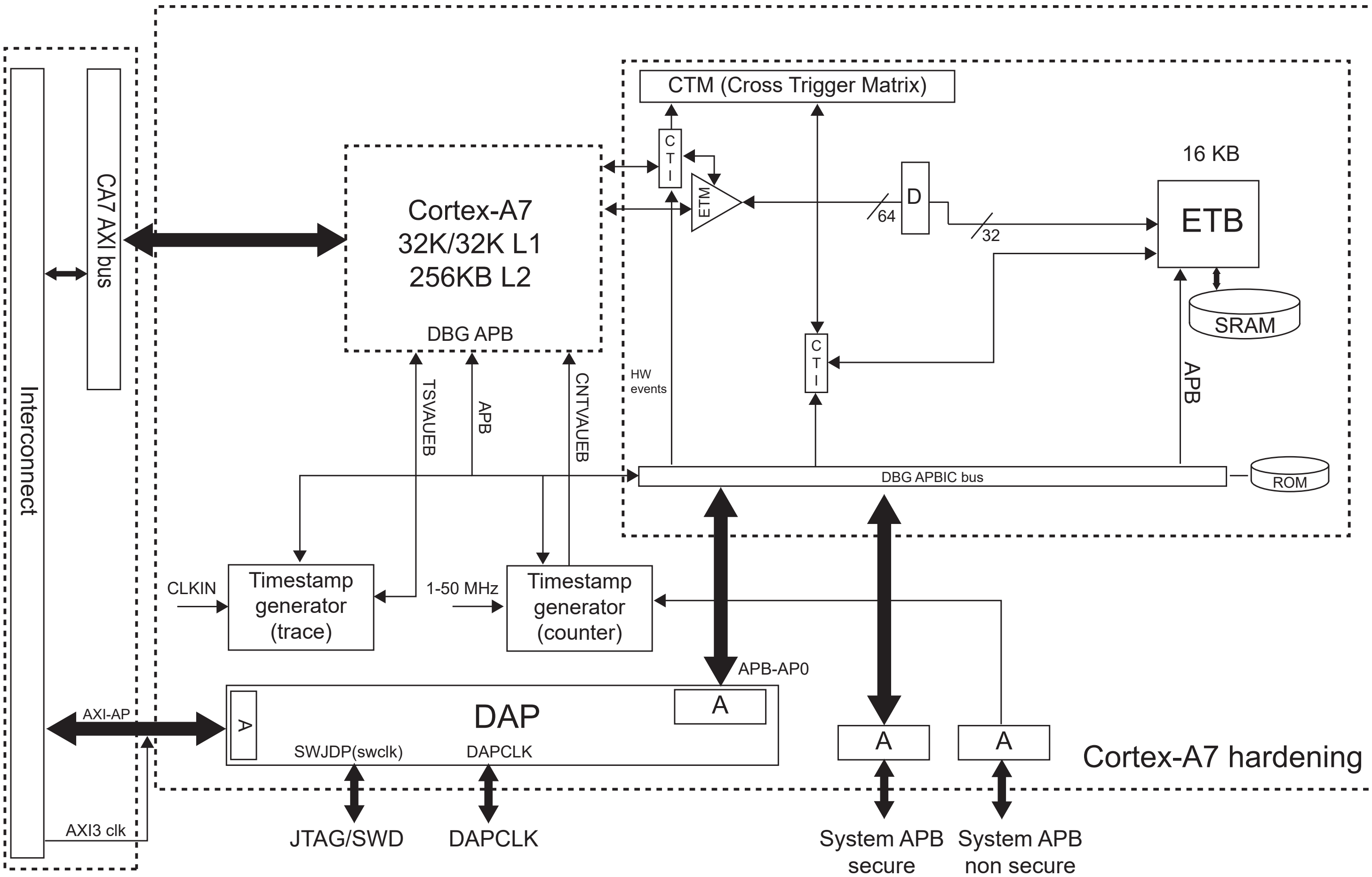
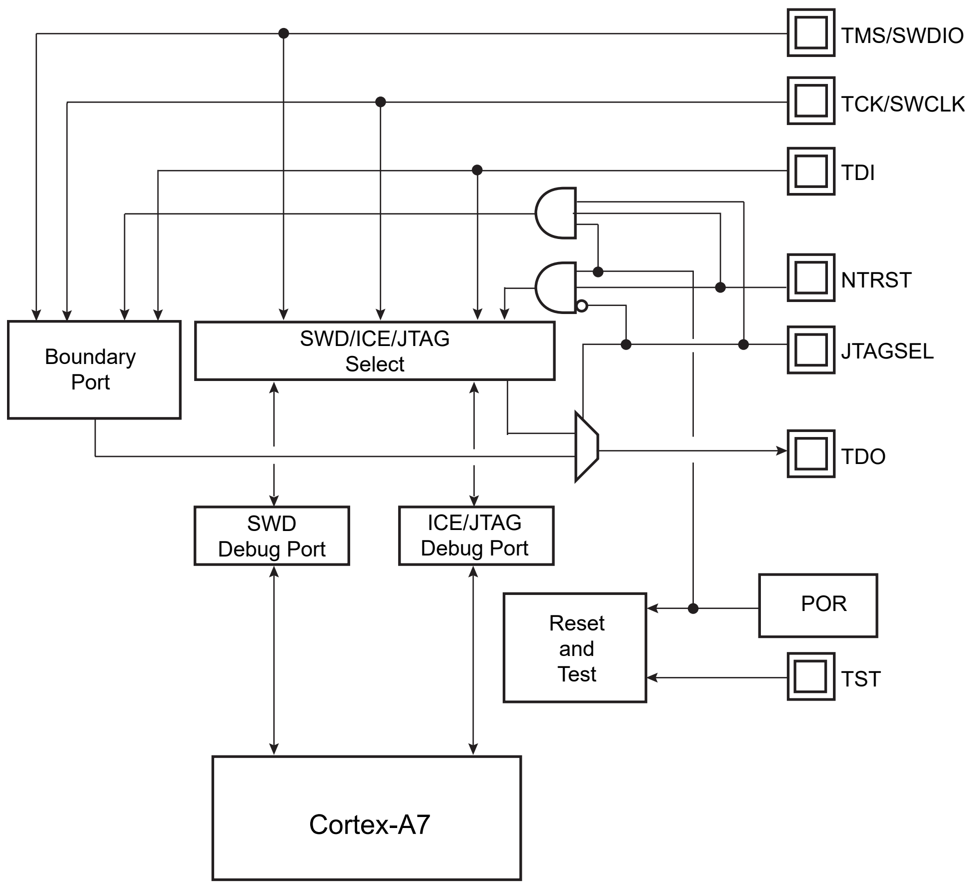


Figure 11-2. Debug and Test Interface Block Diagram



11.4 Pin Description

Table 11-1. Debug and Test Pin List

Pin Name	Function	Type	Active Level
Reset/Test			
TST	Test Mode Select	Input	–
NTRST	Test Reset Signal	Input	–
ICE and JTAG			
TCK/SWCLK	Test Clock/Serial Wire Clock	Input	–
TDI	Test Data In	Input	–
TDO	Test Data Out	Output	–
TMS/SWDIO	Test Mode Select/Serial Wire Input/Output	Input/Output	–
JTAGSEL	JTAG Selection	Input	–

11.5 Functional Description

11.5.1 Test Pin

One dedicated pin, TST, is used to define the device operating mode. The user must make sure that this pin is tied at low level to ensure normal operating conditions. Other values associated with this pin are reserved for manufacturing test.

11.5.2 EmbeddedICE™

The Cortex-A7 EmbeddedICE-RT™ is supported via the ICE/JTAG port. It is connected to a host computer via an ICE interface.The internal state of the Cortex-A7 is examined through an ICE/JTAG port which allows instructions to be serially inserted into the pipeline of the core without using the external data bus. Therefore, when in debug state, a store-multiple (STM) can be inserted into the instruction pipeline. This exports the contents of the Cortex-A7 registers. This data can be serially shifted out without affecting the rest of the system.

There are two scan chains inside the Cortex-A7 processor which support testing, debugging, and programming of the EmbeddedICE-RT. The scan chains are controlled by the ICE/JTAG port.

EmbeddedICE mode is selected when JTAGSEL is low. It is not possible to switch directly between ICE and JTAG operations. A chip reset must be performed after JTAGSEL is changed.

For further details on the EmbeddedICE, refer to the Arm document Cortex-A7 Technical Reference Manual (DDI 0464).

11.5.3 JTAG Signal Description

TMS is the Test Mode Select input which controls the transitions of the test interface state machine.

TDI is the Test Data Input line which supplies the data to the JTAG registers (Boundary Scan Register, Instruction Register, or other data registers).

TDO is the Test Data Output line which is used to serially output the data from the JTAG registers to the equipment controlling the test. It carries the sampled values from the boundary scan chain (or other JTAG registers) and propagates them to the next chip in the serial test circuit.

NTRST (optional in IEEE Standard 1149.1) is a Test-ReSeT input which is mandatory in Arm cores and used to reset the debug logic. On Cortex-A7-based cores, NTRST is a Power-on Reset output. It is asserted on power-on. If necessary, the user can also reset the debug logic with the NTRST pin assertion during 2.5 MCK periods.

TCK is the Test ClocK input which enables the test interface. TCK is pulsed by the equipment controlling the test and not by the tested device. It can be pulsed at any frequency.

11.5.4 Chip Access Using JTAG Connection

The JTAG connection is not enabled by default on this chip at delivery due to the secure ROM code implementation.

By default, the device boots in Standard mode and not in Secure mode. When the secure ROM code starts, it disables the JTAG access for the entire boot sequence.

If the secure ROM code does not find any program in the external memory, it enables the USB connection and the serial port and waits for a dedicated command to switch the chip into Secure mode.

If any other character is received, the secure ROM code starts the standard SAM-BA® monitor, locks access to the ROM memory, and enables the JTAG.

The chip can then be accessed using the JTAG connection.

If the secure ROM code finds a bootable program, it automatically disables ROM access and enables the JTAG connection just before launching the program.

The procedure to enable JTAG access is as follows:

- Connect your computer to the board with JTAG and USB (J20 USB-A).
- Power on the chip.
- Open a terminal console (TeraTerm, HyperTerminal, etc.) on your computer and connect to the USB CDC serial COM port.

- Send the '#' character. You will see then the prompt '>' character sent by the device (indicating that the standard SAM-BA Monitor is running).
- Use the Standard SAM-BA Monitor to connect to the chip with JTAG.

Note that it is not necessary to follow this sequence to connect the standard SAM-BA monitor with USB.

11.5.5 IEEE 1149.1 JTAG Boundary Scan

IEEE 1149.1 JTAG Boundary Scan allows pin-level access independent of the device packaging technology.

IEEE 1149.1 JTAG Boundary Scan is enabled when JTAGSEL is high. The SAMPLE, EXTEST and BYPASS functions are implemented. In ICE debug mode, the Arm processor responds with a non-JTAG chip ID that identifies the processor to the ICE system. This is not IEEE 1149.1 JTAG-compliant.

It is not possible to switch directly between JTAG and ICE operations. A chip reset must be performed after JTAGSEL is changed.

A Boundary-scan Descriptor Language (BSDL) file is provided to set up the test.

11.5.6 JTAG ID Code Register

Name: JTAG ID Code Register
Property: Read-only

JTAG ID Code value is 0x05B4203F.

Bit	31	30	29	28	27	26	25	24
	VERSION[3:0]				PART NUMBER[15:12]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	PART NUMBER[11:4]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	PART NUMBER[3:0]				MANUFACTURER IDENTITY[10:7]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	MANUFACTURER IDENTITY[6:0]							1
Access	R	R	R	R	R	R	R	R
Reset								

Bits 31:28 – VERSION[3:0] Product Version Number
Set to 0x0.

Bits 27:12 – PART NUMBER[15:0] Product Part Number
Product part number is 0x5B42.

Bits 11:1 – MANUFACTURER IDENTITY[10:0]
Set to 0x01f.

Bit 0 – 1
Required by IEEE standard 1149.1. Set to 1.

11.5.7 Cortex-A7 DP Identification Code Register IDCODE

The Identification Code register is always present on all DP implementations. It provides identification information about the Arm Debug Interface.

11.5.7.1 JTAG Debug Port (JTAG-DP)

Name: JTAG Debug Port (JTAG-DP)
Property: Read-only

Debug Port JTAG IDCODE value is 0x6BA00477.
Accessed using its own scan chain, the JTAG-DP Device ID Code Register.

Bit	31	30	29	28	27	26	25	24
	VERSION[3:0]				PART NUMBER[15:12]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	PART NUMBER[11:4]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	PART NUMBER[3:0]				DESIGNER[10:7]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	DESIGNER[6:0]							1
Access	R	R	R	R	R	R	R	R
Reset								

Bits 31:28 – VERSION[3:0] Product Version Number
Set to 0x0.

Bits 27:12 – PART NUMBER[15:0] Product Part Number
Product part number is 0xBA00.

Bits 11:1 – DESIGNER[10:0]
Set to 0x26B.

Bit 0 – 1
Required by IEEE Standard 1149.1. Set to 1.

11.5.7.2 Serial Wire Debug Port (SW-DP)

Name: Serial Wire Debug Port (SW-DP)
Property: Read-only

Debug Port Serial Wire IDCODE is 0x6BA02477.

At address 0x0 on read operations when the APnDP bit = 0. Access to the Identification Code Register is not affected by the value of the CTRLSEL bit in the Select register.

Bit	31	30	29	28	27	26	25	24
	VERSION[3:0]				PART NUMBER[15:12]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	PART NUMBER[11:4]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	PART NUMBER[3:0]				DESIGNER[10:7]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	DESIGNER[6:0]							1
Access	R	R	R	R	R	R	R	R
Reset								

Bits 31:28 – VERSION[3:0] Product Version Number
Set to 0x0.

Bits 27:12 – PART NUMBER[15:0] Product Part Number
Product part number is 0xBA02.

Bits 11:1 – DESIGNER[10:0]
Set to 0x26B.

Bit 0 – 1
Required by IEEE standard 1149.1. Set to 1.

11.5.8 Debug ROM

Table 11-2. Debug ROM

Component	Address (Debugger)	Address (APB System)
Top-level ROM Table (CSSYS)	0x80000000	Base_Ad + 0x00000
ROM Table (PIL)	0x80020000	Base_Ad + 0x20000
Cortex-A7 DBG (PIL)	0x80030000	Base_Ad + 0x30000
Cortex-A7 PMU (PIL)	0x80031000	Base_Ad + 0x31000
CSCTI (PIL)	0x80038000	Base_Ad + 0x38000
ETM-A7 (PIL)	0x8003C000	Base_Ad + 0x3C000
CXETB (CSSYS)	0x80040000	Base_Ad + 0x40000

...........continued

Component	Address (Debugger)	Address (APB System)
CXCTI (CSSYS)	0x80041000	Base_Ad + 0x41000
CXTSGEN Trace (CSSYS)	0x80042000	Base_Ad + 0x42000
CXTSGEN Generic CA-7 Timer (CSSYS)	0x80043000	Base_Ad + 0x43000

12. NIC-400 Global Programmer’s View (NICGPV)

12.1 Description

The system embeds several AXI bus matrixes.

This section describes how to perform priority, arbitration, Quality of Service (QoS) and outstanding configuration settings for AXI peripherals. Refer to the section System Interconnect and Security (SIS) for the description of hosts, clients and interconnections.

12.2 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x2007	Reserved									
0x2008	NICGPV_AMIB_FN_MOD_BM_ISS0	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x200C ... 0x3007	Reserved									
0x3008	NICGPV_AMIB_FN_MOD_BM_ISS1	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x300C ... 0x4007	Reserved									
0x4008	NICGPV_AMIB_FN_MOD_BM_ISS2	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x400C ... 0x5007	Reserved									
0x5008	NICGPV_AMIB_FN_MOD_BM_ISS3	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x500C ... 0x6007	Reserved									
0x6008	NICGPV_AMIB_FN_MOD_BM_ISS4	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x600C ... 0x7007	Reserved									
0x7008	NICGPV_AMIB_FN_MOD_BM_ISS5	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x700C ... 0x8007	Reserved									
0x8008	NICGPV_AMIB_FN_MOD_BM_ISS6	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x800C ... 0x9007	Reserved									
0x9008	NICGPV_AMIB_FN_MOD_BM_ISS7	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x900C ... 0xA007	Reserved									
0xA008	NICGPV_AMIB_FN_MOD_BM_ISS8	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xA00C ... 0xB007	Reserved									
0xB008	NICGPV_AMIB_FN_MOD_BM_ISS9	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xB00C ... 0xC007	Reserved									
0xC008	NICGPV_AMIB_FN_MOD_BM_ISS10	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xC00C ... 0xD007	Reserved									
0xD008	NICGPV_AMIB_FN_MOD_BM_ISS11	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xD00C ... 0xE007	Reserved									
0xE008	NICGPV_AMIB_FN_MOD_BM_ISS12	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xE00C ... 0xF007	Reserved									
0xF008	NICGPV_AMIB_FN_MOD_BM_ISS13	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0xF00C ... 0x010007	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010008	NICGPV_AMIB_FN_MOD_BM_ISS14	31:24								
		23:16								
		15:8								
		7:0							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
0x01000C ... 0x0420FF	Reserved									
0x042100	NICGPV_ASIB_READ_QOS0	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x042104	NICGPV_ASIB_WRITE_QOS0	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x042108 ... 0x04210B	Reserved									
0x04210C	NICGPV_ASIB_QOS_CNTL0	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x042110	NICGPV_ASIB_MAX_OT0	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x042114	NICGPV_ASIB_MAX_COMB_OT0	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x042118	NICGPV_ASIB_AW_P0	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04211C	NICGPV_ASIB_AW_B0	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x042120	NICGPV_ASIB_AW_R0	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x042124	NICGPV_ASIB_AR_P0	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x042128	NICGPV_ASIB_AR_B0	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04212C	NICGPV_ASIB_AR_R0	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x042130	NICGPV_ASIB_TARGET_FC0	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x042134	NICGPV_ASIB_KI_FC0	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x042138	NICGPV_ASIB_QOS_RANGE0	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04213C ... 0x0430FF	Reserved									
0x043100	NICGPV_ASIB_READ_QOS1	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x043104	NICGPV_ASIB_WRITE_QOS1	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x043108 ... 0x04310B	Reserved									
0x04310C	NICGPV_ASIB_QOS_CNTL1	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x043110	NICGPV_ASIB_MAX_OT1	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x043114	NICGPV_ASIB_MAX_COMB_OT1	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x043118	NICGPV_ASIB_AW_P1	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04311C	NICGPV_ASIB_AW_B1	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x043120	NICGPV_ASIB_AW_R1	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x043124	NICGPV_ASIB_AR_P1	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								

..........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x043128	NICGPV_ASIB_AR_B1	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04312C	NICGPV_ASIB_AR_R1	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x043130	NICGPV_ASIB_TARGET_FC1	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x043134	NICGPV_ASIB_KI_FC1	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x043138	NICGPV_ASIB_QOS_RANGE1	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04313C ... 0x0440FF	Reserved									
0x044100	NICGPV_ASIB_READ_QOS2	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x044104	NICGPV_ASIB_WRITE_QOS2	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x044108 ... 0x04410B	Reserved									
0x04410C	NICGPV_ASIB_QOS_CNTL2	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x044110	NICGPV_ASIB_MAX_OT2	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x044114	NICGPV_ASIB_MAX_COMB_OT2	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x044118	NICGPV_ASIB_AW_P2	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04411C	NICGPV_ASIB_AW_B2	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x044120	NICGPV_ASIB_AW_R2	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x044124	NICGPV_ASIB_AR_P2	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x044128	NICGPV_ASIB_AR_B2	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04412C	NICGPV_ASIB_AR_R2	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x044130	NICGPV_ASIB_TARGET_FC2	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x044134	NICGPV_ASIB_KI_FC2	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x044138	NICGPV_ASIB_QOS_RANGE2	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04413C ... 0x0450FF	Reserved									
0x045100	NICGPV_ASIB_READ_QOS3	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x045104	NICGPV_ASIB_WRITE_QOS3	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x045108 ... 0x04510B	Reserved									
0x04510C	NICGPV_ASIB_QOS_CNTL3	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x045110	NICGPV_ASIB_MAX_OT3	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x045114	NICGPV_ASIB_MAX_COMB_OT3	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x045118	NICGPV_ASIB_AW_P3	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04511C	NICGPV_ASIB_AW_B3	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x045120	NICGPV_ASIB_AW_R3	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x045124	NICGPV_ASIB_AR_P3	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x045128	NICGPV_ASIB_AR_B3	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04512C	NICGPV_ASIB_AR_R3	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x045130	NICGPV_ASIB_TARGET_FC3	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x045134	NICGPV_ASIB_KI_FC3	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x045138	NICGPV_ASIB_QOS_RANGE3	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04513C ... 0x0460FF	Reserved									
0x046100	NICGPV_ASIB_READ_QOS4	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x046104	NICGPV_ASIB_WRITE_QOS4	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x046108 ... 0x04610B	Reserved									
0x04610C	NICGPV_ASIB_QOS_CNTL4	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x046110	NICGPV_ASIB_MAX_OT4	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x046114	NICGPV_ASIB_MAX_COMB_OT4	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x046118	NICGPV_ASIB_AW_P4	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04611C	NICGPV_ASIB_AW_B4	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x046120	NICGPV_ASIB_AW_R4	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x046124	NICGPV_ASIB_AR_P4	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x046128	NICGPV_ASIB_AR_B4	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04612C	NICGPV_ASIB_AR_R4	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x046130	NICGPV_ASIB_TARGET_FC4	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x046134	NICGPV_ASIB_KI_FC4	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x046138	NICGPV_ASIB_QOS_RANGE4	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04613C ... 0x0470FF	Reserved									
0x047100	NICGPV_ASIB_READ_QOS5	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x047104	NICGPV_ASIB_WRITE_QOS5	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x047108 ... 0x04710B	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04710C	NICGPV_ASIB_QOS_CNTL5	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x047110	NICGPV_ASIB_MAX_OT5	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x047114	NICGPV_ASIB_MAX_COMB_OT5	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x047118	NICGPV_ASIB_AW_P5	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04711C	NICGPV_ASIB_AW_B5	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x047120	NICGPV_ASIB_AW_R5	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x047124	NICGPV_ASIB_AR_P5	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x047128	NICGPV_ASIB_AR_B5	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04712C	NICGPV_ASIB_AR_R5	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x047130	NICGPV_ASIB_TARGET_FC5	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x047134	NICGPV_ASIB_KI_FC5	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x047138	NICGPV_ASIB_QOS_RANGE5	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04713C ... 0x0480FF	Reserved									
0x048100	NICGPV_ASIB_READ_QOS6	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x048104	NICGPV_ASIB_WRITE_QOS6	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x048108 ... 0x04810B	Reserved									
0x04810C	NICGPV_ASIB_QOS_CNTL6	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x048110	NICGPV_ASIB_MAX_OT6	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x048114	NICGPV_ASIB_MAX_COMB_OT6	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x048118	NICGPV_ASIB_AW_P6	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04811C	NICGPV_ASIB_AW_B6	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x048120	NICGPV_ASIB_AW_R6	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x048124	NICGPV_ASIB_AR_P6	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x048128	NICGPV_ASIB_AR_B6	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04812C	NICGPV_ASIB_AR_R6	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x048130	NICGPV_ASIB_TARGET_FC6	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x048134	NICGPV_ASIB_KI_FC6	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x048138	NICGPV_ASIB_QOS_RANGE6	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04813C ... 0x0490FF	Reserved									
0x049100	NICGPV_ASIB_READ_QOS7	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x049104	NICGPV_ASIB_WRITE_QOS7	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x049108 ... 0x04910B	Reserved									
0x04910C	NICGPV_ASIB_QOS_CNTL7	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x049110	NICGPV_ASIB_MAX_OT7	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x049114	NICGPV_ASIB_MAX_COMB_OT7	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x049118	NICGPV_ASIB_AW_P7	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04911C	NICGPV_ASIB_AW_B7	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x049120	NICGPV_ASIB_AW_R7	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x049124	NICGPV_ASIB_AR_P7	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x049128	NICGPV_ASIB_AR_B7	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04912C	NICGPV_ASIB_AR_R7	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x049130	NICGPV_ASIB_TARGET_FC7	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							

..........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x049134	NICGPV_ASIB_KI_FC7	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x049138	NICGPV_ASIB_QOS_RANGE7	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04913C ... 0x04A0FF	Reserved									
0x04A100	NICGPV_ASIB_READ_QOS8	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x04A104	NICGPV_ASIB_WRITE_QOS8	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x04A108 ... 0x04A10B	Reserved									
0x04A10C	NICGPV_ASIB_QOS_CNTL8	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x04A110	NICGPV_ASIB_MAX_OT8	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x04A114	NICGPV_ASIB_MAX_COMB_OT8	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x04A118	NICGPV_ASIB_AW_P8	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04A11C	NICGPV_ASIB_AW_B8	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x04A120	NICGPV_ASIB_AW_R8	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x04A124	NICGPV_ASIB_AR_P8	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04A128	NICGPV_ASIB_AR_B8	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x04A12C	NICGPV_ASIB_AR_R8	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x04A130	NICGPV_ASIB_TARGET_FC8	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x04A134	NICGPV_ASIB_KI_FC8	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x04A138	NICGPV_ASIB_QOS_RANGE8	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04A13C ... 0x04B0FF	Reserved									
0x04B100	NICGPV_ASIB_READ_QOS9	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x04B104	NICGPV_ASIB_WRITE_QOS9	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x04B108 ... 0x04B10B	Reserved									
0x04B10C	NICGPV_ASIB_QOS_CNTL9	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x04B110	NICGPV_ASIB_MAX_OT9	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x04B114	NICGPV_ASIB_MAX_COMB_OT9	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x04B118	NICGPV_ASIB_AW_P9	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04B11C	NICGPV_ASIB_AW_B9	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x04B120	NICGPV_ASIB_AW_R9	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04B124	NICGPV_ASIB_AR_P9	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04B128	NICGPV_ASIB_AR_B9	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04B12C	NICGPV_ASIB_AR_R9	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x04B130	NICGPV_ASIB_TARGET_FC9	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x04B134	NICGPV_ASIB_KI_FC9	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x04B138	NICGPV_ASIB_QOS_RANGE9	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04B13C ... 0x04C0FF	Reserved									
0x04C100	NICGPV_ASIB_READ_QOS10	31:24								
		23:16								
		15:8								
		7:0					RD_QOS[3:0]			
0x04C104	NICGPV_ASIB_WRITE_QOS10	31:24								
		23:16								
		15:8								
		7:0					WR_QOS[3:0]			
0x04C108 ... 0x04C10B	Reserved									
0x04C10C	NICGPV_ASIB_QOS_CNTL10	31:24								
		23:16				MODE_AR_FC				MODE_AW_FC
		15:8								
		7:0	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
0x04C110	NICGPV_ASIB_MAX_OT10	31:24			AR_MAX_OTI[5:0]					
		23:16	AR_MAX_OTF[7:0]							
		15:8			AW_MAX_OTI[5:0]					
		7:0	AW_MAX_OTF[7:0]							
0x04C114	NICGPV_ASIB_MAX_COMB_OT10	31:24								
		23:16								
		15:8		AWAR_MAX_OTI[6:0]						
		7:0	AWAR_MAX_OTF[7:0]							
0x04C118	NICGPV_ASIB_AW_P10	31:24	AW_P[7:0]							
		23:16								
		15:8								
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04C11C	NICGPV_ASIB_AW_B10	31:24								
		23:16								
		15:8	AW_B[15:8]							
		7:0	AW_B[7:0]							
0x04C120	NICGPV_ASIB_AW_R10	31:24	AW_R[11:4]							
		23:16	AW_R[3:0]							
		15:8								
		7:0								
0x04C124	NICGPV_ASIB_AR_P10	31:24	AR_P[7:0]							
		23:16								
		15:8								
		7:0								
0x04C128	NICGPV_ASIB_AR_B10	31:24								
		23:16								
		15:8	AR_B[15:8]							
		7:0	AR_B[7:0]							
0x04C12C	NICGPV_ASIB_AR_R10	31:24	AR_R[11:4]							
		23:16	AR_R[3:0]							
		15:8								
		7:0								
0x04C130	NICGPV_ASIB_TARGET_FC10	31:24					AR_TGT_LATENCY[11:8]			
		23:16	AR_TGT_LATENCY[7:0]							
		15:8					AW_TGT_LATENCY[11:8]			
		7:0	AW_TGT_LATENCY[7:0]							
0x04C134	NICGPV_ASIB_KI_FC10	31:24								
		23:16								
		15:8						AR_KI[2:0]		
		7:0						AW_KI[2:0]		
0x04C138	NICGPV_ASIB_QOS_RANGE10	31:24					AR_MAX_QOS[3:0]			
		23:16					AR_MIN_QOS[3:0]			
		15:8					AW_MAX_QOS[3:0]			
		7:0					AW_MIN_QOS[3:0]			
0x04C13C ... 0x0C2007	Reserved									
0x0C2008	NICGPV_IB_FN_MOD_BM_ISS0	31:24								
		23:16								
		15:8								
		7:0							FN_MODE[1:0]	
0x0C200C ... 0x0C201F	Reserved									
0x0C2020	NICGPV_IB_SYNC_MODE0	31:24								
		23:16								
		15:8								
		7:0						SYNC_MODE[2:0]		
0x0C2024	NICGPV_IB_FN_MOD2_BP_MRG0	31:24								
		23:16								
		15:8								
		7:0								BP_MRG
0x0C2028 ... 0x0C202B	Reserved									
0x0C202C	NICGPV_IB_FN_MOD_LB0	31:24								
		23:16								
		15:8								
		7:0								LB_MODE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0C2030 ... 0x0C203F	Reserved									
0x0C2040	NICGPV_IB_WR_TIDEMARK0	31:24								
		23:16								
		15:8								
		7:0					WR_TIDEMARK[3:0]			
0x0C2044 ... 0x0C2107	Reserved									
0x0C2108	NICGPV_IB_FN_MOD0	31:24								
		23:16								
		15:8								
		7:0							FN_MODE[1:0]	
0x0C210C ... 0x0C3007	Reserved									
0x0C3008	NICGPV_IB_FN_MOD_BM_ISS1	31:24								
		23:16								
		15:8								
		7:0							FN_MODE[1:0]	
0x0C300C ... 0x0C301F	Reserved									
0x0C3020	NICGPV_IB_SYNC_MODE1	31:24								
		23:16								
		15:8								
		7:0						SYNC_MODE[2:0]		
0x0C3024	NICGPV_IB_FN_MOD2_BP_MRG1	31:24								
		23:16								
		15:8								
		7:0								BP_MRG
0x0C3028 ... 0x0C302B	Reserved									
0x0C302C	NICGPV_IB_FN_MOD_LB1	31:24								
		23:16								
		15:8								
		7:0								LB_MODE
0x0C3030 ... 0x0C303F	Reserved									
0x0C3040	NICGPV_IB_WR_TIDEMARK1	31:24								
		23:16								
		15:8								
		7:0					WR_TIDEMARK[3:0]			
0x0C3044 ... 0x0C3107	Reserved									
0x0C3108	NICGPV_IB_FN_MOD1	31:24								
		23:16								
		15:8								
		7:0							FN_MODE[1:0]	

12.2.1 AMIB Bus Matrix Issuing Functionality Modification Register

Name: NICGPV_AMIB_FN_MOD_BM_ISSx
Offset: 0x2008 + x*0x1000 [x=0..14]
Reset: 0x00000000
Property: Read/Write

This register is only present if the block is connected directly to a switch. This register sets the issuing capability of the preceding switch arbitration scheme to 1.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
Access							R/W	R/W
Reset							0	0

Bit 1 – WRITE_ISS_OVERRIDE Write Issuing Override

Value	Description
0	No effect.
1	Write issuing capability is limited to one outstanding transaction.

Bit 0 – READ_ISS_OVERRIDE Read Issuing Override

Value	Description
0	No effect.
1	Read issuing capability is limited to one outstanding transaction.

12.2.2 ASIB Read Channel QoS Register

Name: NICGPV_ASIB_READ_QOSx
Offset: 0x042100 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is only present when the QoS settings for an ASIB (host) are set to Programmable.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RD_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – RD_QOS[3:0] Read QoS
Read channel QoS value.

12.2.3 ASIB Write Channel QoS Register

Name: NICGPV_ASIB_WRITE_QOSx
Offset: 0x042104 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is only present when the QoS settings for an ASIB (host) are set to Programmable.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					WR_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – WR_QOS[3:0] Write QoS
Write channel QoS value.

12.2.4 ASIB QoS Control Register

Name: NICGPV_ASIB_QOS_CNTLx
Offset: 0x04210C + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register contains the enable bits for all regulators. By default, all bits are set to 0 and no regulation is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				MODE_AR_FC				MODE_AW_FC
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EN_AWAR_OT	EN_AR_OT	EN_AW_OT	EN_AR_FC	EN_AW_FC	EN_AWAR_RATE	EN_AR_RATE	EN_AW_RATE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 20 – MODE_AR_FC AR Feedback Control Mode
Select feedback control regulation for AR of either transaction or address latency.
0 (TRANS_LAT): Transaction latency
1 (ADDR_LAT): Address latency

Bit 16 – MODE_AW_FC AW Feedback Control Mode
Select feedback control regulation for AW of either transaction or address latency.
0 (TRANS_LAT): Transaction latency
1 (ADDR_LAT): Address latency

Bit 7 – EN_AWAR_OT AWAR Outstanding Transactions Enable
0 (DISABLED): Disable combined regulation of outstanding transactions.
1 (ENABLED): Enable combined regulation of outstanding transactions.

Bit 6 – EN_AR_OT AR Outstanding Transactions Enable
0 (DISABLED): Disable regulation of outstanding read transactions.
1 (ENABLED): Enable regulation of outstanding read transactions.

Bit 5 – EN_AW_OT AW Outstanding Transactions Enable
0 (DISABLED): Disable regulation of outstanding write transactions.
1 (ENABLED): Enable regulation of outstanding write transactions.

Bit 4 – EN_AR_FC AR Feedback Control Enable
0 (DISABLED): Disable regulation of AR transaction or address latency.
1 (ENABLED): Enable regulation of AR transaction or address latency using feedback control, depending on the MODE_AR_FC setting.

Bit 3 – EN_AW_FC AW Feedback Control Enable
0 (DISABLED): Disable regulation of AW transaction or address latency.
1 (ENABLED): Enable regulation of AW transaction or address latency using feedback control, depending on the MODE_AW_FC setting.

Bit 2 – EN_AWAR_RATE AW and AR Rates Enable
0 (DISABLED): Disable combined AW and AR rate regulation.
1 (ENABLED): Enable combined AW and AR rate regulation.

Bit 1 – EN_AR_RATE AR Rate Enable
0 (DISABLED): Disable AR rate regulation.
1 (ENABLED): Enable AR rate regulation.

Bit 0 – EN_AW_RATE AW Rate Enable
0 (DISABLED): Disable AW rate regulation.
1 (ENABLED): Enable AW rate regulation.

12.2.5 ASIB Maximum Number of Outstanding Transactions Register

Name: NICGPV_ASIB_MAX_OTx
Offset: 0x042110 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program the maximum number of address requests for the AR and AW channels.

Bit	31	30	29	28	27	26	25	24
			AR_MAX_OTI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	AR_MAX_OTF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
			AW_MAX_OTI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	AW_MAX_OTF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – AR_MAX_OTI[5:0] AR Maximum Outstanding Transaction Integer
Corresponds to the integer part of the maximum outstanding AR addresses.

Bits 23:16 – AR_MAX_OTF[7:0] AR Maximum Outstanding Transaction Fraction
Corresponds to the fractional part of the maximum outstanding AR addresses.

Bits 13:8 – AW_MAX_OTI[5:0] AW Maximum Outstanding Transaction Integer
Corresponds to the integer part of the maximum outstanding AW addresses.

Bits 7:0 – AW_MAX_OTF[7:0] AW Maximum Outstanding Transaction Fraction
Corresponds to the fractional part of the maximum outstanding AW addresses.

12.2.6 ASIB Maximum Combined Outstanding Transactions Register

Name: NICGPV_ASIB_MAX_COMB_OTx
Offset: 0x042114 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program the maximum number of address requests for the AR and AW channels.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		AWAR_MAX_OTI[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AWAR_MAX_OTF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 14:8 – AWAR_MAX_OTI[6:0] AW and AR Combined Maximum Outstanding Transactions Integer
Integer part of maximum combined outstanding AW and AR addresses

Bits 7:0 – AWAR_MAX_OTF[7:0] AW and AR Combined Maximum Outstanding Transactions Fraction
Fractional part of maximum combined outstanding AW and AR addresses

12.2.7 ASIB Write Address Channel Peak Rate Register

Name: NICGPV_ASIB_AW_Px
Offset: 0x042118 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a binary fraction of the peak number of transfers per cycle.

Bit	31	30	29	28	27	26	25	24
	AW_P[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Access								
Reset								

Bits 31:24 – AW_P[7:0] AW Channel Peak Rate

12.2.8 ASIB Write Address Channel Burstiness Allowance Register

Name: NICGPV_ASIB_AW_Bx
Offset: 0x04211C + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a burstiness allowance, in number of transfers.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	AW_B[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AW_B[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – AW_B[15:0] AW Channel Burstiness

12.2.9 ASIB Write Address Channel Average Rate Register

Name: NICGPV_ASIB_AW_Rx
Offset: 0x042120 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a binary fraction of the average number of transfers per cycle.

Bit	31	30	29	28	27	26	25	24
	AW_R[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AW_R[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:20 – AW_R[11:0] AW Channel Average Rate

12.2.10 ASIB Read Address Channel Peak Rate Register

Name: NICGPV_ASIB_AR_Px
Offset: 0x042124 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a binary fraction of the peak number of transfers per cycle.

Bit	31	30	29	28	27	26	25	24
	AR_P[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:24 – AR_P[7:0] AR Channel Peak Rate

12.2.11 ASIB Read Address Channel Burstiness Allowance Register

Name: NICGPV_ASIB_AR_Bx
Offset: 0x042128 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a burstiness allowance, in number of transfers.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	AR_B[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	AR_B[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – AR_B[15:0] AR Channel Burstiness

12.2.12 ASIB Read Address Channel Average Rate Register

Name: NICGPV_ASIB_AR_Rx
Offset: 0x04212C + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a binary fraction of the average number of transfers per cycle.

Bit	31	30	29	28	27	26	25	24
	AR_R[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AR_R[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:20 – AR_R[11:0] AR Channel Average Rate

12.2.13 ASIB Feedback Controlled Target Register

Name: NICGPV_ASIB_TARGET_FCx
Offset: 0x042130 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a target latency, in cycles, for the regulation of reads and writes.

Bit	31	30	29	28	27	26	25	24
					AR_TGT_LATENCY[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	AR_TGT_LATENCY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					AW_TGT_LATENCY[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	AW_TGT_LATENCY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – AR_TGT_LATENCY[11:0] AR Channel Target Latency

Bits 11:0 – AW_TGT_LATENCY[11:0] AW Channel Target Latency

12.2.14 ASIB Feedback Controlled Scale Register

Name: NICGPV_ASIB_KI_FCx
Offset: 0x042134 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program a latency regulation value, awqos or arqos, a scale factor coded for powers of 2 in the 2^{-3} to 2^{-10} range, to match a 16-bit integrator.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						AR_KI[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						AW_KI[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – AR_KI[2:0] AW QOS Scale Factor
Power of 2 in the 2^{-3} to 2^{-10} range

Value	Name	Description
0	TWO_POW_M3	2^{-3}
1	TWO_POW_M4	2^{-4}
2	TWO_POW_M5	2^{-5}
3	TWO_POW_M6	2^{-6}
4	TWO_POW_M7	2^{-7}
5	TWO_POW_M8	2^{-8}
6	TWO_POW_M9	2^{-9}
7	TWO_POW_M10	2^{-10}

Bits 2:0 – AW_KI[2:0] AR QOS Scale Factor
Power of 2 in the 2^{-3} to 2^{-10} range

Value	Name	Description
0	TWO_POW_M3	2^{-3}
1	TWO_POW_M4	2^{-4}
2	TWO_POW_M5	2^{-5}
3	TWO_POW_M6	2^{-6}
4	TWO_POW_M7	2^{-7}
5	TWO_POW_M8	2^{-8}
6	TWO_POW_M9	2^{-9}
7	TWO_POW_M10	2^{-10}

12.2.15 ASIB QoS Range Register

Name: NICGPV_ASIB_QOS_RANGEx
Offset: 0x042138 + x*0x1000 [x=0..10]
Reset: 0x00000000
Property: Read/Write

This register is used to program the minimum and maximum values for the arqos and awqos signals generated by the latency regulators.

Bit	31	30	29	28	27	26	25	24
					AR_MAX_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
					AR_MIN_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
					AW_MAX_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
					AW_MIN_QOS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – AR_MAX_QOS[3:0] AR Maximum QOS
Maximum arqos

Bits 19:16 – AR_MIN_QOS[3:0] AR Minimum QOS
Minimum arqos

Bits 11:8 – AW_MAX_QOS[3:0] AW Maximum QOS
Maximum awqos

Bits 3:0 – AW_MIN_QOS[3:0] AW Minimum QOS
Minimum awqos

12.2.16 IB Bus Matrix Issuing Functionality Modification Register

Name: NICGPV_IB_FN_MOD_BM_ISSx
Offset: 0x0C2008 + x*0x1000 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register sets the issuing capability of the preceding switch arbitration scheme to 1.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							FN_MODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – FN_MODE[1:0] Issuing Functionality
The register bits are active high.

Value	Name	Description
0	READ_ISS_OVERRIDE	Read issuing.
1	WRITE_ISS_OVERRIDE	Write issuing.

12.2.17 IB Clock Boundary Synchronization Scheme Register

Name: NICGPV_IB_SYNC_MODEx
Offset: 0x0C2020 + x*0x1000 [x=0..1]
Reset: 0x00000004
Property: Read/Write

This register is only present if a programmable clock crossing is configured across the interface bus.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYNC_MODE[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	0

Bits 2:0 – SYNC_MODE[2:0] Read Issuing
Selects the required clock boundary synchronization scheme by programming the register value.

Value	Name	Description
0	SYNC_1_TO_1	sync 1:1
1	SYNC_M_TO_1	sync m:1
2	SYNC_1_TO_N	sync 1:n
3	SYNC_M_TO_N	sync m:n
4	ASYNC	async
5	RESERVED	Reserved
6	RESERVED	Reserved
7	RESERVED	Reserved

12.2.18 IB Bypass Merge Register

Name: NICGPV_IB_FN_MOD2_BP_MRGx
Offset: 0x0C2024 + x*0x1000 [x=0..1]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								BP_MRG
Access								R/W
Reset								0

Bit 0 – BP_MRG Bypass Merge

12.2.19 IB Long Burst Functionality Modification Register

Name: NICGPV_IB_FN_MOD_LBx
Offset: 0x0C202C + x*0x1000 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register is only present when downsizing to AXI4 is required.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								LB_MODE
Access								R/W
Reset								0

Bit 0 – LB_MODE Long Burst Mode

Value	Name	Description
0	DISABLED	Long bursts cannot be generated at the output of the interface bus.
1	ENABLED	Long bursts can be generated at the output of the interface bus.

12.2.20 IB Tidemark Register

Name: NICGPV_IB_WR_TIDEMARKx
Offset: 0x0C2040 + x*0x1000 [x=0..1]
Reset: 0x00000002
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					WR_TIDEMARK[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bits 3:0 – WR_TIDEMARK[3:0] WR Tidemark Value

12.2.21 IB Issuing Functionality Modification Register

Name: NICGPV_IB_FN_MODx
Offset: 0x0C2108 + x*0x1000 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register sets the block issuing capability to one outstanding transaction.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							FN_MODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – FN_MODE[1:0] Issuing Functionality Mode
The register bits are active high.

Value	Name	Description
0	READ_ISS_OVERRIDE	Read issuing.
1	WRITE_ISS_OVERRIDE	Write issuing.

13. Bus Matrix (MATRIX)

13.1 Description

The system embeds one AHB bus matrix.

This section describes how to perform priority, arbitration and security settings for peripherals indicated as “MATRIX” in the table “System Interconnections”. These include memories (USB_RAM, SRAM_P0, SRAM_P1, SMC, QSPI0, QSPI1, NFC_RAM) and ISC Host.

Refer to the section System Interconnect and Security (SIS) for the description of hosts, clients and interconnections.

13.2 Embedded Characteristics

- Support for Long Bursts (32, 64, 128 and 256-bit long)
- Enhanced Programmable Mixed Arbitration for Each Client
  - Round-robin
  - Fixed priority
  - Latency Quality of Service
- Programmable Default Host for Each Client
  - No default host
  - Last accessed default host
  - Fixed default host
- Deterministic Maximum Access Latency for Hosts
- Zero or One Cycle Arbitration Latency for the First Access of a Burst
- Bus Lock Forwarding to Clients
- Host Number Forwarding to Clients
- One Special Function Register for Each Client (Not dedicated)
- Register Write Protection of User Interface Registers
- Arm TrustZone Technology

13.3 Memory Mapping

The MATRIX provides one decoder for every host interface. The decoder offers each host several memory mappings. Each memory area can be assigned to several clients. Booting at the same address while using different clients (i.e., external RAM, internal ROM or internal Flash, etc.) becomes possible.

13.4 Special Bus Granting Techniques

The MATRIX provides some speculative bus granting techniques in order to anticipate access requests from hosts. Hence, latency is reduced at first access of a burst or, for a single transfer, as long as the client is free from any other host access. It does not provide any benefit if the client is continuously accessed by more than one host, since arbitration is pipelined and has no negative effect on the client bandwidth or access latency.

This bus granting technique sets a different default host for every client.

At the end of the current access, if no other request is pending, the client remains connected to its associated default host. A client can be associated with three kinds of default hosts:

- no default host
- last access host
- fixed default host

To change from one type of default host to another, the user interface provides Client Configuration registers (MATRIX_SCFGx), one for every client, which set a default host for each client. MATRIX_SCFGx contain two fields to manage host selection: DEFMSTR_TYPE and FIXED_DEFMSTR. The 2-bit DEFMSTR_TYPE field selects the default host type (no default, last access host, fixed default host), whereas the 4-bit FIXED_DEFMSTR field selects a fixed default host provided that DEFMSTR_TYPE is set to fixed default host. Refer to section 13.11.2. MATRIX_SCFGx.

13.5 No Default Host

After the end of the current access, if no other request is pending, the client is disconnected from all hosts.

This configuration incurs one latency clock cycle for the first access of a burst after bus Idle. Arbitration without default host may be used for hosts that perform significant bursts or several transfers with no Idle cycle in-between, or if the client bus bandwidth is widely used by one or more hosts.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput whatever the number of requesting hosts.

13.6 Last Access Host

After the end of the current access, if no other request is pending, the client remains connected to the last host that performed an access request.

This enables the MATRIX to remove the one latency cycle for the last host that accessed the client. Other non-privileged hosts still get one latency clock cycle if they need to access the same client. This technique is useful for hosts that mainly perform single accesses or short bursts with some Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput whatever is the number of requesting hosts.

13.7 Fixed Default Host

After the end of the current access, if no other request is pending, the client connects to its fixed default host. Unlike the last access host, the fixed default host does not change unless the user modifies it by software (FIXED_DEFMSTR field of the related MATRIX_SCFG).

This allows the MATRIX arbiters to remove the one latency clock cycle for the fixed default host of the client. All requests attempted by the fixed default host do not cause any arbitration latency, whereas other non-privileged hosts get one latency cycle. This technique is useful for a host that mainly performs single accesses or short bursts with Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput, regardless of the number of requesting hosts.

13.8 Arbitration

The MATRIX provides an arbitration technique that reduces latency when conflicts occur, i.e., when two or more hosts try to access the same client at the same time. One arbiter per client is provided, thus arbitrating each client specifically.

The user can either choose one of the following arbitration types, or mix them for each client:

1. Round-robin Arbitration (default)
2. Fixed Priority Arbitration

The resulting algorithm may be complemented by selecting a default host configuration for each client.

When re-arbitration must be done, specific conditions apply. See section 13.8.1. Arbitration Scheduling .

13.8.1 Arbitration Scheduling

Each arbiter has the ability to arbitrate between two or more different host requests. In order to avoid burst breaking as well as to provide the maximum throughput for client interfaces, arbitration may only take place during the following cycles:

- Idle Cycles: When a client is not connected to any host or is connected to a host which is not currently accessing it.
- Single Cycles: When a client is currently doing a single access.
- End of Burst Cycles: When the current cycle is the last cycle of a burst transfer. For defined length burst, predicted end of burst matches the size of the transfer but is managed differently for undefined length burst. See section 13.8.1.1. Undefined Length Burst Arbitration.
- Slot Cycle Limit: When the slot cycle counter has reached the limit value indicating that the current host access is too long and must be broken. See section 13.8.1.2. Slot Cycle Limit Arbitration.

13.8.1.1 Undefined Length Burst Arbitration

In order to prevent long burst lengths that can lock the access to the client for an excessive period of time, the user can trigger the re-arbitration before the end of the incremental bursts. The re-arbitration period can be selected from the following Undefined Length Burst Type (ULBT) possibilities:

- Unlimited: no predetermined end of burst is generated. This value enables 1 Kbyte burst lengths.
- 1-beat bursts: predetermined end of burst is generated at each single transfer during the INCR transfer.
- 4-beat bursts: predetermined end of burst is generated at the end of each 4-beat boundary during INCR transfer.
- 8-beat bursts: predetermined end of burst is generated at the end of each 8-beat boundary during INCR transfer.
- 16-beat bursts: predetermined end of burst is generated at the end of each 16-beat boundary during INCR transfer.
- 32-beat bursts: predetermined end of burst is generated at the end of each 32-beat boundary during INCR transfer.
- 64-beat bursts: predetermined end of burst is generated at the end of each 64-beat boundary during INCR transfer.
- 128-beat bursts: predetermined end of burst is generated at the end of each 128-beat boundary during INCR transfer.

Undefined-length bursts lower than 8 beats should not be used since this may decrease the overall bus bandwidth due to arbitration and client latencies at each first access of a burst.

However, if the length of undefined-length bursts is known for a host, it is recommended to configure MATRIX_MCFG.ULBT accordingly.

13.8.1.2 Slot Cycle Limit Arbitration

The MATRIX contains specific logic to break long accesses, such as very long bursts on a very slow client (e.g., an external low speed memory). At each arbitration time, a counter is loaded with the value previously written in MATRIX_SCFGx.SLOT_CYCLE and decreased at each clock cycle. When the counter elapses, the arbiter has the ability to re-arbitrate at the end of the current system bus access cycle.

Unless a host has a very tight access latency constraint, which could lead to data overflow or underflow due to a badly undersized internal FIFO with respect to its throughput, the Slot Cycle Limit should be disabled (SLOT_CYCLE = 0) or set to its default maximum value in order not to inefficiently break long bursts performed by some hosts.

In most cases, this feature is not needed and should be disabled for power saving.

⚠WARNING This feature cannot prevent any client from locking its access indefinitely.

13.8.2 Arbitration Priority Scheme

The MATRIX arbitration scheme is organized in priority pools, each corresponding to an access criticality class as shown in the “Latency Quality of Service” column in the table below. When the Latency Quality of Service is enabled for a host-client pair through the MATRIX, the priority pool number to use for arbitration at the client port is determined from the host. When the Latency Quality of Service is disabled, it is determined through the MATRIX user interface. See 13.11.3. MATRIX_PRASx.

After reset, the Latency Quality of Service is enabled by default on all of the host ports that are connected to a host driving the Latency Quality of Service signals, as shown in the bit LQOSEN of 13.11.3. MATRIX_PRASx and 13.11.4. MATRIX_PRBSx.

Table 13-1. Arbitration Priority Pools

Priority Pool	Latency Quality of Service
3	Latency Critical
2	Latency Sensitive
1	Bandwidth Sensitive
0	Background Transfers

Round-robin priority is used in the highest and lowest priority pools 3 and 0, whereas fixed level priority is used between priority pools and in the intermediate priority pools 2 and 1.

For each client, each host is assigned to one of the client priority pools based on the Latency Quality of Service inputs or to the priority registers for clients (MxPR fields of MATRIX_PRAS and MATRIX_PRBS). When evaluating host requests, this priority pool level always takes precedence.

After reset, most of the hosts belong to the lowest priority pool (MxPR = 0, Background Transfer) and are therefore granted bus access in a true round-robin order.

The highest priority pool must be specifically reserved for hosts requiring very low access latency. If more than one host belongs to this pool, those hosts are granted bus access in a biased round-robin manner which enables tight and deterministic maximum access latency from system bus requests. In the worst case, any currently occurring high-priority host request is granted after the current bus host access has ended and any other high priority pool host requests have been granted once each.

The lowest priority pool shares the remaining bus bandwidth between hosts.

Intermediate priority pools enable fine priority tuning. Typically, a latency-sensitive host or a bandwidth-sensitive host use such a priority level. The higher the priority level (MxPR value), the higher the host priority.

For good CPU performance, it is recommended to let the CPU priority configured with the default reset value 2 (Latency Sensitive).

All combinations of MxPR values are allowed for all hosts and clients. For example, some hosts might be assigned the highest priority pool (round-robin), and remaining hosts the lowest priority pool (round-robin), with no host for intermediate fixed priority levels.

13.8.2.1 Fixed Priority Arbitration

The fixed priority arbitration algorithm is the first and only arbitration algorithm applied between hosts from distinct priority pools. It is also used in priority pools other than the highest and lowest priority pools (intermediate priority pools).

Fixed priority arbitration enables the MATRIX arbiters to dispatch the requests from different hosts to the same client by using the fixed priority defined by the user in the MxPR field for each host in the registers, MATRIX_PRAS and MATRIX_PRBS. If two or more host requests are active at the same time, the host with the highest priority MxPR number is serviced first.

In intermediate priority pools, if two or more host requests with the same priority are active at the same time, the host with the highest number is serviced first.

13.8.2.2 Round-Robin Arbitration

This algorithm is only used in the highest and lowest priority pools. It allows the MATRIX arbiters to properly dispatch requests from different hosts to the same client. If two or more host requests are active at the same time in the priority pool, they are serviced in a round-robin increasing host number order.

13.9 Register Write Protection

To prevent any single software error from corrupting MATRIX behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the Write Protection Mode register (MATRIX_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status register (MATRIX_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is reset by writing the MATRIX_WPMR with the appropriate access key WPKEY.

The registers listed below can be write-protected.

Related Links

13.11.1. MATRIX_MCFGx
13.11.2. MATRIX_SCFGx
13.11.3. MATRIX_PRASx
13.11.4. MATRIX_PRBSx
13.11.5. MATRIX_MRCR
13.11.6. MATRIX_MEIER
13.11.7. MATRIX_MEIDR
13.11.13. MATRIX_SSRx
13.11.14. MATRIX_SASSRx
13.11.15. MATRIX_SRTSRx
13.11.16. MATRIX_SPSELRx

13.10 TrustZone Extension

TrustZone secure software is supported through the filtering of each client access with host security bit extension signals.

TrustZone technology adds the ability to manage the access rights for Secure and Non-Secure accesses. The access rights are defined through the hardware and software configuration of the device. The operating mode is the following:

- Hosts transmit requests with the secure or non-secure Security option.
- The MATRIX, according to its configuration and the request, grants or denies the access.

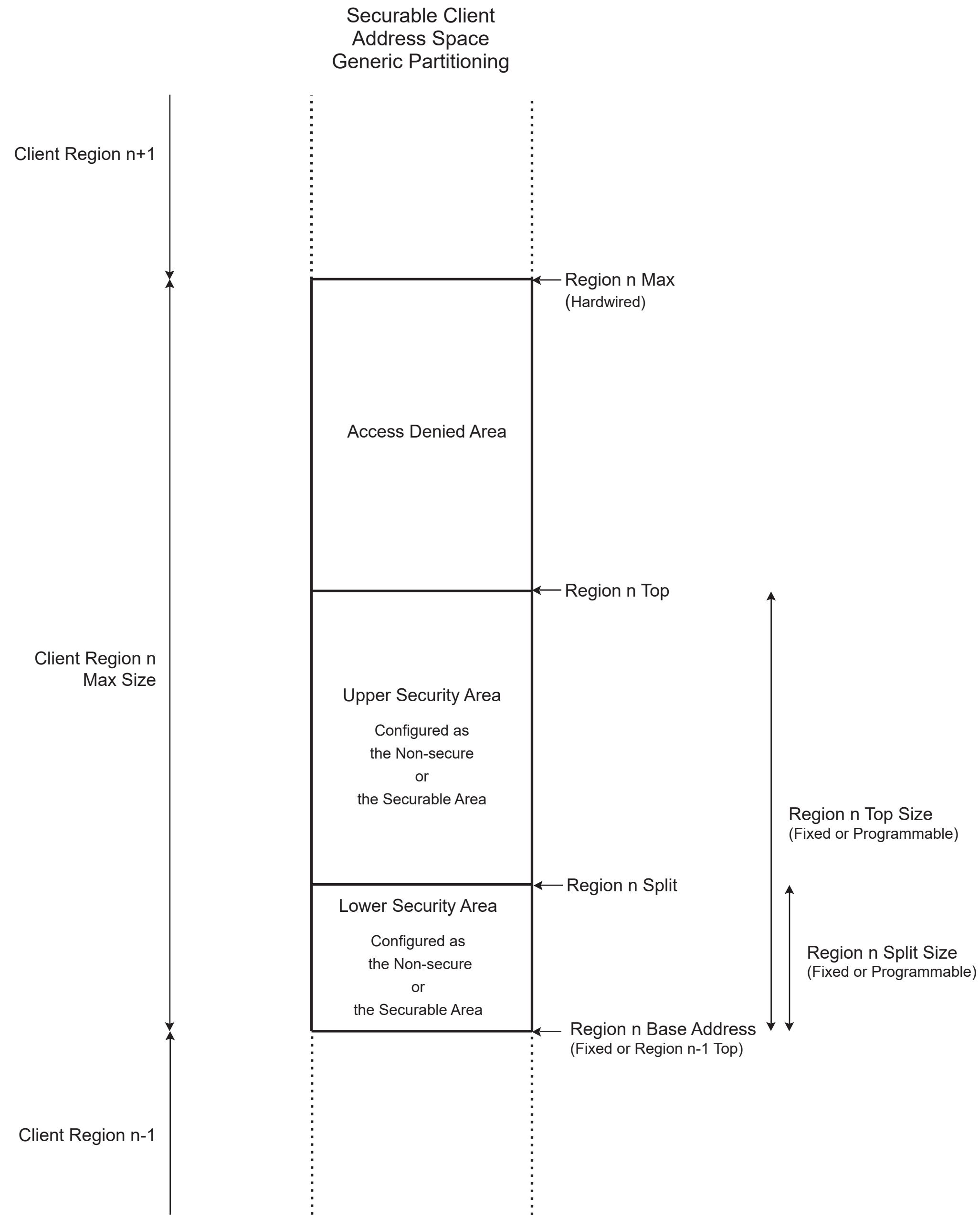
The client address space is divided into one or more client regions. The client regions are generally contiguous parts of the client address space. The client region is potentially split into an access denied area (upper part) and a security region which can be split (lower part), unless the client security region occupies the whole client region. The security region itself can be split into one secure area and one non-secure area. The secure area may be independently secured for read access and for write access.

For one client region, the following characteristics are configured by hardware or software:

- Base Address of the client region
- Max Size of the client region—a maximum size for the region's physical content
- Top Size of the client security region— the actually programmed or fixed size for the region's physical content
- Split Size of the client security region— the size of one of the two security areas of the region

The following figure shows how the terms defined here are implemented in a client address space.

Figure 13-1. Generic Partitioning of the Client Address Space



A set of security registers specifies, for each client, the client security region or client security area, the security mode required to access this client, client security region or client security area. See section 13.11.13. MATRIX_SSRx, 13.11.14. MATRIX_SASSRx and 13.11.15. MATRIX_SRTSRx.

These registers can only be accessed in Secure mode.

The MATRIX propagates the security bit down to the clients to let them perform additional security checks, and the MATRIX itself allows or denies the access to the clients by means of its TrustZone embedded controller.

Access violations may be reported either by a client through the bus error response, or by the embedded TrustZone controller. In both cases, a bus error response is sent to the offending host and the error is flagged in the Host Error Status Register. An interrupt can be sent to the Secure world, if it has been enabled for that host by writing into the

Host Error Interrupt Enable Register. Thus, the offending host is identified. The offending address is registered in the Host Error Address Registers, so that the client and the targeted security region are also known.

Depending on the hardware parameters and software configuration, the address space of each client security region may or may not be split into two parts, one belonging to the Secure world and the other one to the Normal world.

Five different security types of clients are supported. The number of security regions is set by design for each client, independently, from 1 to 8, totaling from 1 up to 16 security areas for each security-configurable client.

13.10.1 Security Types of System Bus Clients

13.10.1.1 Principles

The MATRIX supports five different security types of clients: two fixed types and three configurable types. The security type of a client is set at hardware design among the following:

- Never Secure
- Always Secure
- Internal Securable
- External Securable
- Scalable Securable

The security type is set at hardware design on a per-host and a per-client basis. Never Secure and Always Secure security types are not software configurable.

The different security types have the following characteristics:

- **Never Secure** clients have no security mode access restriction. Their address space is precisely set by design. Any out-of-address range access is denied and reported.
- **Always Secure** clients can only be accessed by a secure host request. Their address space is precisely set by design. Any non-secure or out-of-address range access is denied and reported.
- **Internal Securable** is intended for internal RAM. The Internal Securable client has one client region which has a hardware fixed base address and Security Region Top. This client region may be split through software configuration into one Non-secure area plus one Secure area. Inside the client security region, the split boundary is programmable in powers of 2 from 4 Kbytes up to the full client security region address space. The security area located below the split boundary may be configured as the Non-secure or the Secure one. The Securable area may be independently configured as Read Secured and/or Write Secured. Any access with security or address range violation is denied and reported.
- **External Securable** is intended for external memories on the EBI, such as DDR, SDRAM, external ROM or NAND Flash. The External Securable client has identical features as the Internal Securable client, plus the ability to configure each of its client security region address space sizes according to the external memory parts used. This avoids mirroring Secure areas into Non-secure areas, and further restricts the overall accessible address range. Any access with security or configured address range violation is denied and reported.
- **Scalable Securable** is intended for external memories with a dedicated client, such as DDR. The Scalable Securable client is divided into a fixed number of scalable, equally sized, and contiguous security regions. Each of them can be split in the same way as for Internal or External Securable clients. The security region size must be configured by software, so that the equally-sized regions fill the actual available memory. This avoids mirroring Secure areas into Non-secure areas, and further restricts the overall accessible address range. Any access with security or configured address range violation is denied and reported.

As the security type is set at hardware design on a per-host and per-client basis, it is possible to set some client access security as configurable from one or some particular hosts, and to set the access as Always Secure from all the other hosts.

As the security type is set by design at the client region level, different security region types can be mixed inside a single client.

Likewise, the mapping base address and the accessible address range of each client or client region may have been hardware-restricted on a per-host basis from no access to full client address space.

13.10.1.2 Examples

The table below shows an example of Security Type settings.

Table 13-2. Security Type Setting Example

Client	Host0	Host1	Host2
Client0 Internal Memory	Always Not Secured	Internal Securable 1 Region	Internal Securable 1 Region
Client1 EBI	External Securable 2 Regions	Always Secured	External Securable 2 Regions

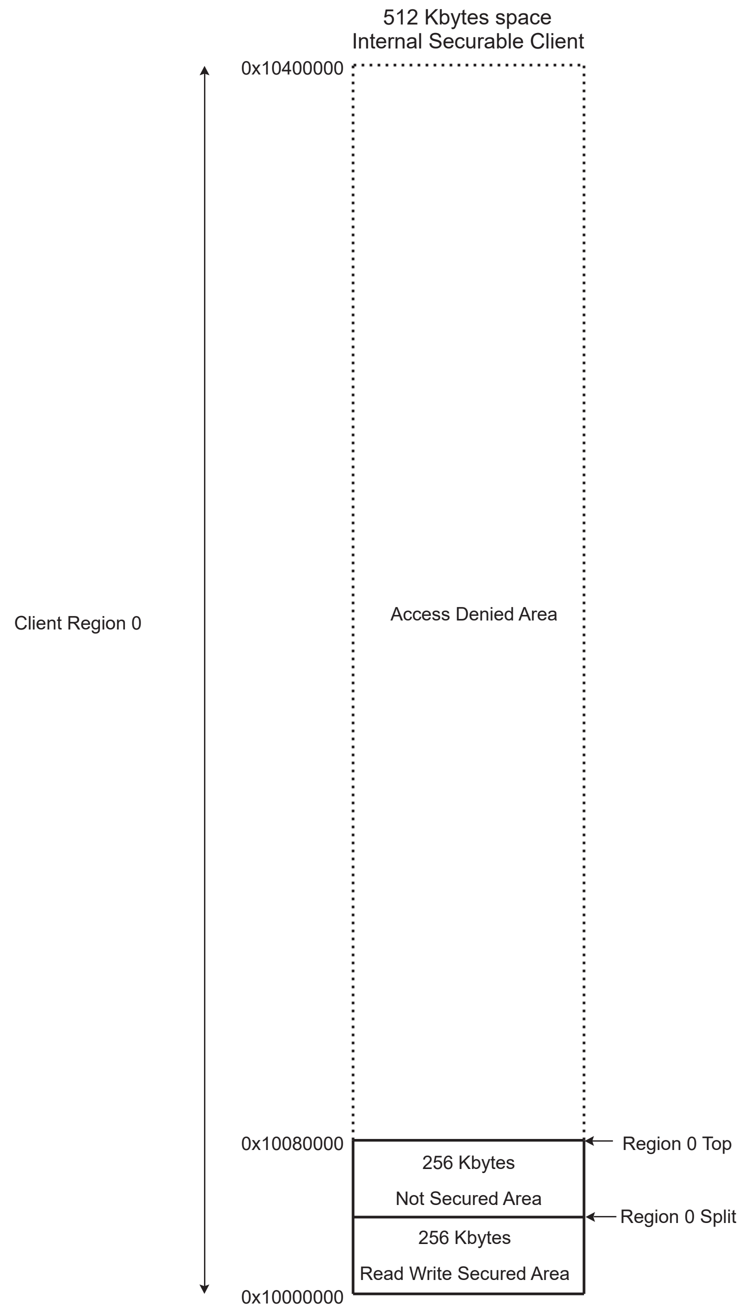
This example is constructed with the following characteristics:

- Client0 is an Internal Memory containing one region:
  - The Access from Host0 to Client0 is Never Secure
  - The access from Host1 and Host2 to Client0 is Internal Securable with one region and with the same software configuration (Choice of SASPLIT0 and the security configuration bits LANSECH, RDNSECH, WRNSECH).
- Client1 is an EBI containing two regions:
  - The access from Host1 to Client1 is Always Secure
  - The access from Host0 and Host2 to Client1 is External Securable with two regions and with the same software configuration (Choice of SRTOP0, SRTOP1, SASPLIT0, SASPLIT1 and the security configuration bits LANSECH, RDNSECH, WRNSECH).

The following figure shows an Internal Securable client example. This example is constructed with the following hypothesis:

- The client is an Internal Memory containing one region. The Client region Max Size is 4 Mbytes.
- The client region 0 base address equals 0x10000000. Its Top Size is 512 Kbytes (hardware configuration).
- The client software configuration is:
  - SASPLIT0 is set to 256 Kbytes.
  - LANSECH0 is set to 0, the low area of region 0 is the Securable one.
  - RDNSECH0 is set to 0, region 0 Securable area is secured for reads.
  - WRNSECH0 is set to 0, region 0 Securable area is secured for writes.

Figure 13-2. Partitioning Example of an Internal Securable Client Featuring 1 Security Region of 512 Kbytes Split into 1 or 2 Security Areas of 4 Kbytes to 512 Kbytes



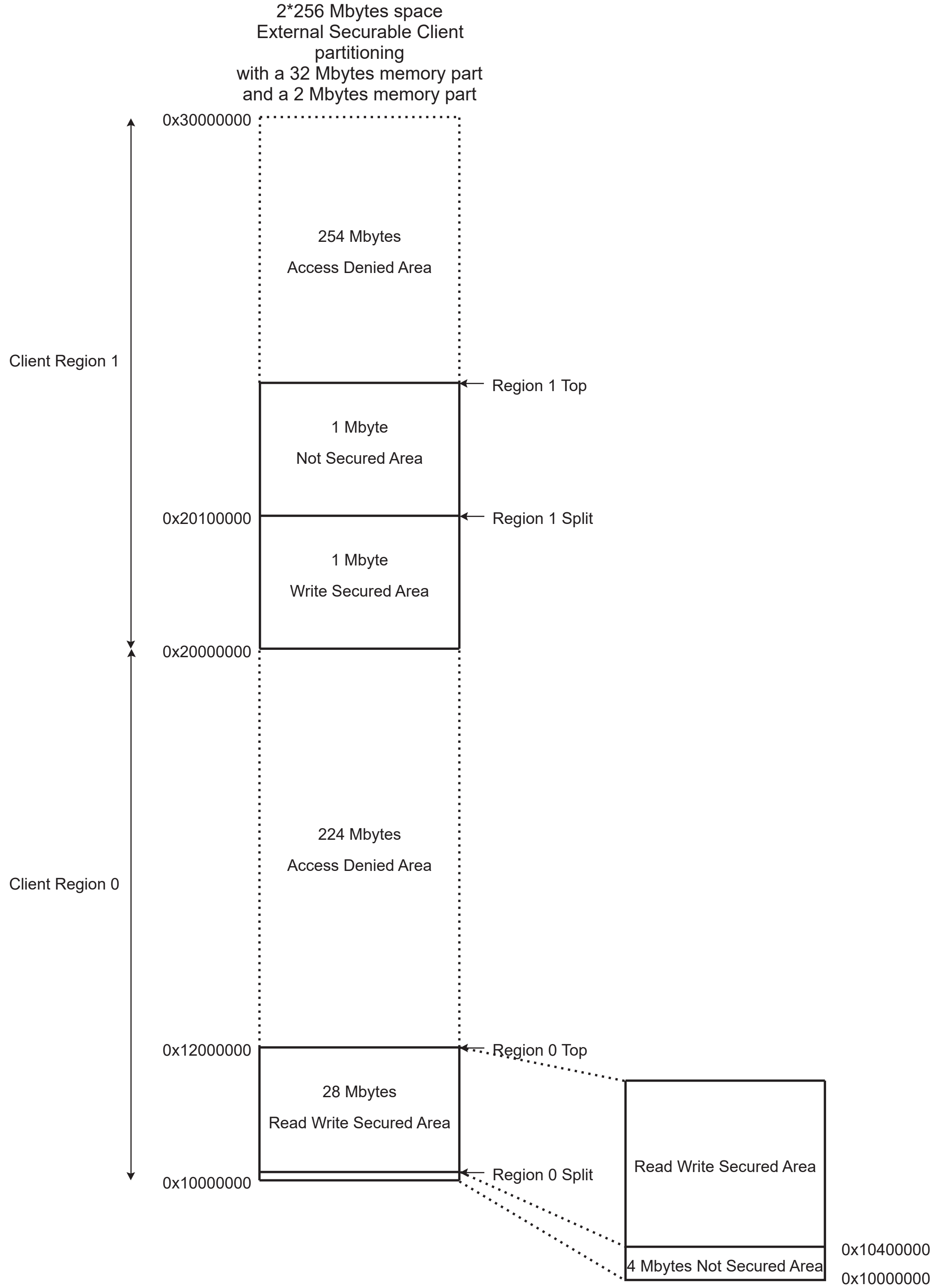
Note: The client security areas split inside the security region are configured by writing into the Security Areas Split Client Registers.

The following figure shows an External Securable client example. This example is constructed with the following hypothesis:

- The client is an interface with the external bus (EBI) containing two regions. The client size is 2 ´ 256 Mbytes. Each client region max size is 256 Mbytes.
- The client region 0 base address equals 0x10000000. It is connected to a 32 Mbyte memory, for example an external DDR. The client region 0 top size must be set to 32 Mbytes.
- The client region 1 base address equals 0x20000000. It is connected to a 2 Mbyte memory, for example an external NAND Flash. The client region 1 top size must be set to 2 Mbytes.

- The client software configuration is:
  - SRTOP0 is set to 32 Mbytes
  - SRTOP1 is set to 2 Mbytes
  - SASPLIT0 is set to 4 Mbytes
  - SASPLIT1 is set to 1 Mbyte
  - LANSECH0 is set to 1, the low area of region 0 is the non-Securable one
  - RDNSECH0 is set to 0, region 0 Securable area is secured for reads
  - WRNSECH0 is set to 0, region 0 Securable area is secured for writes
  - LANSECH1 is set to 0, the low area of region 1 is the Securable one
  - RDNSECH1 is set to 1, region 1 Securable area is non-secured for reads
  - WRNSECH1 is set to 0, region 1 Securable area is secured for writes

Figure 13-3. Partitioning Example: External Securable Client Featuring 2 Security Regions of 4 Kbytes to 128 Mbytes Each and Up to 4 Security Areas of 4 Kbytes to 128 Mbytes

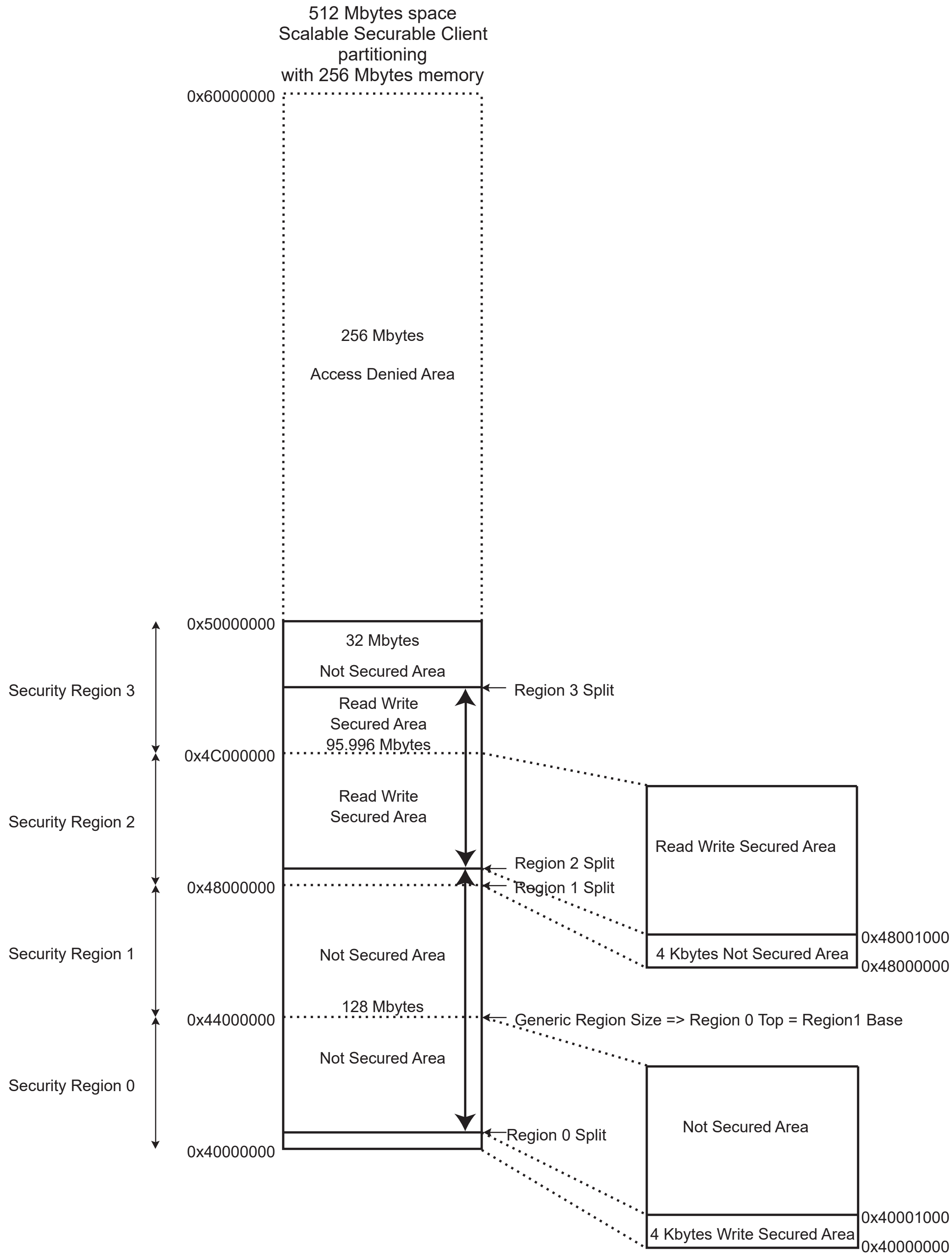


Note: The client region sizes are configured by writing into the Security Region Top Client Registers. The client security area split inside each region is configured by writing into the Security Areas Split Client Registers.

The following figure shows a Scalable Securable client example. This example is constructed with the following hypothesis:

- The client is an external memory with dedicated client containing four regions, for example an external DDR.
- The client size is 512 Mbytes.
- The client base address equals 0x40000000. It is connected to a 256-Mbyte external memory.
- As the connected memory size is 256 Mbytes and there are four regions, the size of each region is 64 Mbytes. This gives the value of the client region max size and top size. The client region 0 Top Size must be configured to 64 Mbytes.
- The client software configuration is as follows:
  – SRTOP0 is set to 64 Mbytes
  – SASPLIT0 is set to 4 Kbytes
  – SASPLIT1 is set to 64 Mbytes, so its low area occupies the whole region 1
  – SASPLIT2 is set to 4 Kbytes
  – SASPLIT3 is set to 32 Mbytes
  – LANSECH0 is set to 0, the low area of region 0 is the Securable one
  – RDNSECH0 is set to 1, region 0 Securable area is not secured for reads.
  – WRNSECH0 is set to 0, region 0 Securable area is secured for writes.
  – LANSECH1 is set to 1, the low area of region 1 is the non-securable one.
  – RDNSECH1 is ‘don’t care’ since the low area occupies the whole region 1.
  – WRNSECH1 is ‘don’t care’ since the low area occupies the whole region 1.
  – LANSECH2 is set to 1, the low area of region 2 is the non-Securable one.
  – RDNSECH2 is set to 0, region 2 Securable area is secured for reads
  – WRNSECH2 is set to 0, region 2 Securable area is secured for writes
  – LANSECH3 is set to 0, the low area of region 3 is the Securable one.
  – RDNSECH3 is set to 0, region 3 Securable area is secured for reads.
  – WRNSECH3 is set to 0, region 3 Securable area is secured for writes.

Figure 13-4. Partitioning Example: Scalable Securable Client Featuring 4 Equally-sized Security Regions of 1 Mbytes to 128 Mbytes Each and Up to 8 Security Areas of 4 Kbytes to 128 Mbytes



Note: The clients' generic security regions sizes are configured by writing into field SRTOP0 of the Security Region Top Client Registers. The custom client security areas split inside each region are configured by writing into the Security Areas Split Client Registers.

13.10.2 Security Types of System Bus Hosts

Hosts send requests to the MATRIX with a security attribute that depends on the host security type, which is identical to the security type of the client user interface.

For DMA, the TrustZone security attribute can be selected for each channel. Refer to the section “Extensible DMA Controller (XDMAC)”.

13.10.3 Security of Peripheral Bus Clients

To configure the security mode required for accessing a peripheral bus client connected behind the system-to-peripheral bus (AHB/APB Bridge), the MATRIX features three 32-bit Security Peripheral Select x Registers. Some of these bits may have been set by design to an Always Secure or a Never Secure value, when others are programmed by software (see 13.11.16. MATRIX_SPSELRx).

The system-to-peripheral bus bridge compares the incoming host request security bit with the required security mode for the selected peripheral, and accepts or denies access. In the last case, its bus error response is internally flagged in the Host Error Status Register; the offending address is registered in the Host Error Address Registers.

13.11 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MATRIX_MCFG0	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
...										
0x38	MATRIX_MCFG14	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
0x3C ... 0x3F	Reserved									
0x40	MATRIX_SCFG0	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x44	MATRIX_SCFG1	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x48	MATRIX_SCFG2	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x4C	MATRIX_SCFG3	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x50	MATRIX_SCFG4	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x54	MATRIX_SCFG5	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x58	MATRIX_SCFG6	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x5C	MATRIX_SCFG7	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x60	MATRIX_SCFG8	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x64	MATRIX_SCFG9	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x68 ... 0x7F	Reserved									
0x80	MATRIX_PRAS0	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x84	MATRIX_PRBS0	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x88	MATRIX_PRAS1	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x8C	MATRIX_PRBS1	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x90	MATRIX_PRAS2	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x94	MATRIX_PRBS2	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x98	MATRIX_PRAS3	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x9C	MATRIX_PRBS3	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA0	MATRIX_PRAS4	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xA4	MATRIX_PRBS4	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA8	MATRIX_PRAS5	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xAC	MATRIX_PRBS5	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB0	MATRIX_PRAS6	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xB4	MATRIX_PRBS6	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB8	MATRIX_PRAS7	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xBC	MATRIX_PRBS7	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC0	MATRIX_PRAS8	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xC4	MATRIX_PRBS8	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC8	MATRIX_PRAS9	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xCC	MATRIX_PRBS9	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xD0 ... 0xFF	Reserved									
0x0100	MATRIX_MRCR	31:24								
		23:16								
		15:8		RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
		7:0	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
0x0104 ... 0x014F	Reserved									
0x0150	MATRIX_MEIER	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0154	MATRIX_MEIDR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0158	MATRIX_MEIMR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x015C	MATRIX_MESR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0160	MATRIX_MEAR0	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
...										
0x0198	MATRIX_MEAR14	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
0x019C ... 0x01E3	Reserved									
0x01E4	MATRIX_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	CFGFRZ							WPEN
0x01E8	MATRIX_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x01EC ... 0x01FF	Reserved									
0x0200	MATRIX_SSR0	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0204	MATRIX_SSR1	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0208	MATRIX_SSR2	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x020C	MATRIX_SSR3	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0210	MATRIX_SSR4	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0214	MATRIX_SSR5	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0218	MATRIX_SSR6	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x021C	MATRIX_SSR7	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0220	MATRIX_SSR8	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0224	MATRIX_SSR9	31:24	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
		23:16	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
		15:8	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
		7:0	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
0x0228 ... 0x023F	Reserved									
0x0240	MATRIX_SASSR0	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0244	MATRIX_SASSR1	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0248	MATRIX_SASSR2	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x024C	MATRIX_SASSR3	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0250	MATRIX_SASSR4	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0254	MATRIX_SASSR5	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0258	MATRIX_SASSR6	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x025C	MATRIX_SASSR7	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0260	MATRIX_SASSR8	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0264	MATRIX_SASSR9	31:24	SASPLIT7[3:0]				SASPLIT6[3:0]			
		23:16	SASPLIT5[3:0]				SASPLIT4[3:0]			
		15:8	SASPLIT3[3:0]				SASPLIT2[3:0]			
		7:0	SASPLIT1[3:0]				SASPLIT0[3:0]			
0x0268 ... 0x027F	Reserved									
0x0280	MATRIX_SRTSR0	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0284	MATRIX_SRTSR1	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x0288	MATRIX_SRTSR2	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x028C	MATRIX_SRTSR3	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x0290	MATRIX_SRTSR4	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x0294	MATRIX_SRTSR5	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x0298	MATRIX_SRTSR6	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x029C	MATRIX_SRTSR7	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x02A0	MATRIX_SRTSR8	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x02A4	MATRIX_SRTSR9	31:24	SRTOP7[3:0]				SRTOP6[3:0]			
		23:16	SRTOP5[3:0]				SRTOP4[3:0]			
		15:8	SRTOP3[3:0]				SRTOP2[3:0]			
		7:0	SRTOP1[3:0]				SRTOP0[3:0]			
0x02A8 ... 0x02BF	Reserved									
0x02C0	MATRIX_SPSELR1	31:24	NSECP31	NSECP30	NSECP29	NSECP28	NSECP27	NSECP26	NSECP25	NSECP24
		23:16	NSECP23	NSECP22	NSECP21	NSECP20	NSECP19	NSECP18	NSECP17	NSECP16
		15:8	NSECP15	NSECP14	NSECP13	NSECP12	NSECP11	NSECP10	NSECP9	NSECP8
		7:0	NSECP7	NSECP6	NSECP5	NSECP4	NSECP3	NSECP2	NSECP1	NSECP0
0x02C4	MATRIX_SPSELR2	31:24	NSECP31	NSECP30	NSECP29	NSECP28	NSECP27	NSECP26	NSECP25	NSECP24
		23:16	NSECP23	NSECP22	NSECP21	NSECP20	NSECP19	NSECP18	NSECP17	NSECP16
		15:8	NSECP15	NSECP14	NSECP13	NSECP12	NSECP11	NSECP10	NSECP9	NSECP8
		7:0	NSECP7	NSECP6	NSECP5	NSECP4	NSECP3	NSECP2	NSECP1	NSECP0
0x02C8	MATRIX_SPSELR3	31:24	NSECP31	NSECP30	NSECP29	NSECP28	NSECP27	NSECP26	NSECP25	NSECP24
		23:16	NSECP23	NSECP22	NSECP21	NSECP20	NSECP19	NSECP18	NSECP17	NSECP16
		15:8	NSECP15	NSECP14	NSECP13	NSECP12	NSECP11	NSECP10	NSECP9	NSECP8
		7:0	NSECP7	NSECP6	NSECP5	NSECP4	NSECP3	NSECP2	NSECP1	NSECP0

13.11.1 MATRIX Host Configuration Register x

Name: MATRIX_MCFGx
Offset: 0x00 + x*0x04 [x=0..14]
Reset: 0x00000004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ULBT[2:0]	
Access						R/W	R/W	R/W
Reset						1	0	0

Bits 2:0 – ULBT[2:0] Undefined Length Burst Type

Value	Name	Description
0	UNLIMITED	Unlimited Length Burst—No predicted end of burst is generated, therefore INCR bursts coming from this host can only be broken if the Client Slot Cycle Limit is reached. If the Slot Cycle Limit is not reached, the burst is normally completed by the host, at the latest, on the next system bus 1 Kbyte address boundary, allowing up to 256-beat word bursts or 128-beat double-word bursts. This value should not be used in the very particular case of a host capable of performing back-to-back undefined length bursts on a single client, since this could indefinitely freeze the client arbitration and thus prevent another host from accessing this client.
1	SINGLE	Single Access—The undefined length burst is treated as a succession of single accesses, allowing re-arbitration at each beat of the INCR burst or bursts sequence.
2	4_BEAT	4-beat Burst—The undefined length burst or bursts sequence is split into 4-beat bursts or less, allowing re-arbitration every 4 beats.
3	8_BEAT	8-beat Burst—The undefined length burst or bursts sequence is split into 8-beat bursts or less, allowing re-arbitration every 8 beats.
4	16_BEAT	16-beat Burst—The undefined length burst or bursts sequence is split into 16-beat bursts or less, allowing re-arbitration every 16 beats.
5	32_BEAT	32-beat Burst—The undefined length burst or bursts sequence is split into 32-beat bursts or less, allowing re-arbitration every 32 beats.
6	64_BEAT	64-beat Burst—The undefined length burst or bursts sequence is split into 64-beat bursts or less, allowing re-arbitration every 64 beats.
7	128_BEAT	128-beat Burst—The undefined length burst or bursts sequence is split into 128-beat bursts or less, allowing re-arbitration every 128 beats. Unless duly needed, the ULBT should be left at its default 0 value for power saving.

13.11.2 MATRIX Client Configuration Register x

Name: MATRIX_SCFGx
Offset: 0x40 + x*0x04 [x=0..9]
Reset: 0x000001FF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

–

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
								SLOT_CYCLE[8]
Access								R/W
Reset								1

Bit	7	6	5	4	3	2	1	0
	SLOT_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 21:18 – FIXED_DEFMSTR[3:0] Fixed Default Host
This is the number of the Default Host for this client. Only used if DEFMSTR_TYPE is 2. Specifying the number of a host which is not connected to the selected client is equivalent to setting DEFMSTR_TYPE to 0.

Bits 17:16 – DEFMSTR_TYPE[1:0] Default Host Type

Value	Name	Description
0	NONE	No Default Host—At the end of the current client access, if no other host request is pending, the client is disconnected from all hosts. This results in a one clock cycle latency for the first access of a burst transfer or for a single access.
1	LAST	Last Default Host—At the end of the current client access, if no other host request is pending, the client stays connected to the last host having accessed it. This results in not having one clock cycle latency when the last host tries to access the client again.
2	FIXED	Fixed Default Host—At the end of the current client access, if no other host request is pending, the client connects to the fixed host the number that has been written in the FIXED_DEFMSTR field. This results in not having one clock cycle latency when the fixed host tries to access the client again.

Bits 8:0 – SLOT_CYCLE[8:0] Maximum Bus Grant Duration for Hosts
When SLOT_CYCLE system bus clock cycles have elapsed since the last arbitration, a new arbitration takes place to let another host access this client. If another host is requesting the client bus, then the current host burst is broken.

If SLOT_CYCLE = 0, the Slot Cycle Limit feature is disabled and bursts always complete unless broken according to the ULBT.
This limit has been placed in order to enforce arbitration so as to meet potential latency constraints of hosts waiting for client access.
This limit must not be too small. Unreasonably small values break every burst and the MATRIX arbitrates without performing any data transfer. The default maximum value is usually an optimal conservative choice.
In most cases, this feature is not needed and should be disabled for power saving.
See section Slot Cycle Limit Arbitration for details.

13.11.3 MATRIX Priority Register A For Clients x

Name: MATRIX_PRASx
Offset: 0x80 + x*0x08 [x=0..9]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 2, 6, 10, 14, 18, 22, 26, 30 – LQOSENx Latency Quality of Service Enable for Host x

Value	Description
0	Disables propagation of Latency Quality of Service from the Host x to the Client and apply MxPR priority for all access from Host x to the Client.
1	Enables the propagation of Latency Quality of Service from the Host x to the Client if supported by the Host x.

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25, 28:29 – MxPR Host x Priority
Fixed priority of Host x for accessing the selected client. The higher the number, the higher the priority.
All the hosts programmed with the same MxPR value for the client make up a priority pool.
Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.
Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).
See section Arbitration Priority Scheme for details.
If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the client.
If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Host x.
For hosts other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU host, the usual value of this field should be 0x2.

13.11.4 MATRIX Priority Register B For Clients x

Name: MATRIX_PRBSx
Offset: 0x84 + x*0x08 [x=0..9]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						LQOSEN14	M14PR[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 2, 6, 10, 14, 18, 22, 26 – LQOSENx Latency Quality of Service Enable for Host x

Value	Description
0	Disables propagation of Latency Quality of Service from the Host x to the Client and apply MxPR priority for all access from Host x to the Client.
1	Enables the propagation of Latency Quality of Service from the Host x to the Client if supported by the Host x.

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25 – MxPR Host x Priority
Fixed priority of Host x for accessing the selected client. The higher the number, the higher the priority.
All the hosts programmed with the same MxPR value for the client make up a priority pool.
Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.
Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).
See section Arbitration Priority Scheme for details.
If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the client.
If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Host x.
For hosts other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU host, the usual value of this field should be 0x2.

13.11.5 MATRIX Host Remap Control Register

Name: MATRIX_MRCR
Offset: 0x0100
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – RCBx Remap Command Bit for Host x

Value	Description
0	Disables remapped address decoding for the selected host.
1	Enables remapped address decoding for the selected host.

13.11.6 MATRIX Host Error Interrupt Enable Register

Name: MATRIX_MEIER
Offset: 0x0150
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Host x Access Error

Value	Description
0	No effect.
1	Enables Host x Access Error interrupt source.

13.11.7 MATRIX Host Error Interrupt Disable Register

Name: MATRIX_MEIDR
Offset: 0x0154
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Host x Access Error

Value	Description
0	No effect.
1	Disables Host x Access Error interrupt source.

13.11.8 MATRIX Host Error Interrupt Mask Register

Name: MATRIX_MEIMR
Offset: 0x0158
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Host x Access Error

Value	Description
0	Host x Access Error does not trigger any interrupt.
1	Host x Access Error triggers the MATRIX interrupt line.

13.11.9 MATRIX Host Error Status Register

Name: MATRIX_MESR
Offset: 0x015C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Host x Access Error

Value	Description
0	No Host Access Error has occurred since the last read of the MATRIX_MESR.
1	At least one Host Access Error has occurred since the last read of the MATRIX_MESR.

13.11.10 MATRIX Host Error Address Register x

Name: MATRIX_MEARx
Offset: 0x0160 + x*0x04 [x=0..14]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ERRADD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ERRADD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ERRADD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ERRADD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ERRADD[31:0] Host Error Address
32 most significant bits of the last access error address

13.11.11 MATRIX Write Protection Mode Register

Name: MATRIX_WPMR
Offset: 0x01E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFGFRZ							WPEN
Access	R/W							R/W
Reset	0							0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4D4154	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and CFGFRZ bits. Always reads as 0.

Bit 7 – CFGFRZ Configuration Freeze

Value	Description
0	The MATRIX configuration is not frozen.
1	Freezes the MATRIX configuration until hardware reset. The registers that can be protected by the WPEN bit and the Write Protection Mode Register are no longer modifiable.

Bit 0 – WPEN Write Protection Enable
See section Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).

13.11.12 MATRIX Write Protection Status Register

Name: MATRIX_WPSR
Offset: 0x01E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last write of the MATRIX_WPMR.
1	A write protection violation has occurred since the last write of the MATRIX_WPMR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

13.11.13 MATRIX Security Client Register x

Name: MATRIX_SSRx
Offset: 0x0200 + x*0x04 [x=0..9]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DSSOA7	DSSOA6	DSSOA5	DSSOA4	DSSOA3	DSSOA2	DSSOA1	DSSOA0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WRNSECH7	WRNSECH6	WRNSECH5	WRNSECH4	WRNSECH3	WRNSECH2	WRNSECH1	WRNSECH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RDNSECH7	RDNSECH6	RDNSECH5	RDNSECH4	RDNSECH3	RDNSECH2	RDNSECH1	RDNSECH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LANSECH7	LANSECH6	LANSECH5	LANSECH4	LANSECH3	LANSECH2	LANSECH1	LANSECH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – DSSOAx Downward Security Split Offset Address for HSELx Security Region

Value	Description
0	For the HSELx client security region, the security area split MATRIX_SASSR / SASPLITx defines the size of the client security area starting from the base address of the client security region and up.
1	For the HSELx client security region, the security area split MATRIX_SASSR / SASPLITx defines the size of the client security area starting from the end address of the client security region and down.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – WRNSECHx Write Not Secured for HSELx Security Region
Securable Area access rights:

WRNSECHx / RDNSECHx	Non-Secure Access	Secure Access
00	Denied	Write - Read
01	Read	Write - Read
10	Write	Write - Read
11	Write - Read	Write - Read

Value	Description
0	The HSELx client security region is split into one Write Secured and one Write Not Secured area, according to LANSECHx and MATRIX_SASSR / SASPLITx. That is, the so defined securable high or low area is Secured for Write access.
1	The HSELx client security region is Not Secured for Write access.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – RDNSECHx Read Not Secured for HSELx Security Region

Value	Description
0	The HSELx client security region is split into one Read Secured and one Read Not Secured area, according to LANSECHx and MATRIX_SASSR / SASPLITx. That is, the so defined securable high or low area is Secured for Read access.
1	The HSELx client security region is Not Secured for Read access.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – LANSECHx Low Area Not Secured in HSELx Security Region

Value	Description
0	The security of the HSELx client area laying below the corresponding MATRIX_SASSR / SASPLITx boundary is configured according to RDNSECHx and WRNSECHx. The whole remaining HSELx upper address space is configured as Not Secured access.
1	The HSELx client address area laying below the corresponding MATRIX_SASSR / SASPLITx boundary is configured as Not Secured access, and the whole remaining upper address space according to RDNSECHx and WRNSECHx.

13.11.14 MATRIX Security Areas Split Client Register x

Name: MATRIX_SASSRx
Offset: 0x0240 + x*0x04 [x=0..9]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	SASPLIT7[3:0]				SASPLIT6[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SASPLIT5[3:0]				SASPLIT4[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SASPLIT3[3:0]				SASPLIT2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SASPLIT1[3:0]				SASPLIT0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – SASPLITx Security Areas Split for HSELx Security Region
This field defines the boundary address offset where the HSELx client security region splits into two Security Areas whose access is controlled according to the corresponding MATRIX_SSR.
If this size is set at or above the HSELx Region Size, then the MATRIX_SSR settings apply to the unique security area then covering the whole HSELx Security Region.

SASPLITx	Split Offset	Security Area Size
0000	0x00001000	4 Kbytes
0001	0x00002000	8 Kbytes
0010	0x00004000	16 Kbytes
0011	0x00008000	32 Kbytes
0100	0x00010000	64 Kbytes
0101	0x00020000	128 Kbytes
0110	0x00040000	256 Kbytes
0111	0x00080000	512 Kbytes
1000	0x00100000	1 Mbyte
1001	0x00200000	2 Mbytes
1010	0x00400000	4 Mbytes
1011	0x00800000	8 Mbytes
1100	0x01000000	16 Mbytes
1101	0x02000000	32 Mbytes
1110	0x04000000	64 Mbytes
1111	0x08000000	128 Mbytes

13.11.15 MATRIX Security Region Top Client Register x

Name: MATRIX_SRTSRx
Offset: 0x0280 + x*0x04 [x=0..9]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	SRTOP7[3:0]				SRTOP6[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SRTOP5[3:0]				SRTOP4[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SRTOP3[3:0]				SRTOP2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SRTOP1[3:0]				SRTOP0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – SRTOPx HSELx Security Region Top
This field defines the size of the HSELx security region address space. Invalid sizes for the client region must never be programmed. Valid sizes and number of security regions are product, client and client configuration dependent.
Note: The clients featuring multiple scalable contiguous security regions have a single SRTOP0 field for all the security regions.

If this HSELx security region size is set at or below the MATRIX_SASSR / SASPLITx Security Areas Split size, then the MATRIX_SSR settings apply to the unique security area then covering the whole HSELx security region

SRTOPx	Top Offset	Security Region Size
0000	0x00001000	4 Kbytes
0001	0x00002000	8 Kbytes
0010	0x00004000	16 Kbytes
0011	0x00008000	32 Kbytes
0100	0x00010000	64 Kbytes
0101	0x00020000	128 Kbytes
0110	0x00040000	256 Kbytes
0111	0x00080000	512 Kbytes
1000	0x00100000	1 Mbyte
1001	0x00200000	2 Mbytes
1010	0x00400000	4 Mbytes
1011	0x00800000	8 Mbytes
1100	0x01000000	16 Mbytes
1101	0x02000000	32 Mbytes
1110	0x04000000	64 Mbytes
1111	0x08000000	128 Mbytes

13.11.16 MATRIX Security Peripheral Select x Register [x=1..3]

Name: MATRIX_SPSELRx
Offset: 0x02C0 + (x-1)*0x04 [x=1..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Each MATRIX_SPSELR can configure the access security type for up to 32 peripherals:

- MATRIX_SPSELR1 configures the access security type for peripheral identifiers 0–31 (bits NSECP0–NSECP31).
- MATRIX_SPSELR2 configures the access security type for peripheral identifiers 32–63 (bits NSECP0–NSECP31).
- MATRIX_SPSELR3 configures the access security type for peripheral identifiers 64–95 (bits NSECP0–NSECP31).

Values in the MATRIX Security registers are product-dependent.

The actual number of peripherals implemented is device-specific; refer to section “Peripheral Identifiers” for details.

Bit	31	30	29	28	27	26	25	24
	NSECP31	NSECP30	NSECP29	NSECP28	NSECP27	NSECP26	NSECP25	NSECP24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NSECP23	NSECP22	NSECP21	NSECP20	NSECP19	NSECP18	NSECP17	NSECP16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NSECP15	NSECP14	NSECP13	NSECP12	NSECP11	NSECP10	NSECP9	NSECP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NSECP7	NSECP6	NSECP5	NSECP4	NSECP3	NSECP2	NSECP1	NSECP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NSECPy Not Secured Peripheral

Value	Description
0	The selected peripheral address space is configured as “Secured” access (value of ‘0’ has no effect if the peripheral security type is “Not Secured”).
1	The selected peripheral address space is configured as “Not Secured” access (value of ‘1’ has no effect if the peripheral security type is “Always Secured”).

14. DMA Controller (XDMAC)

14.1 Description

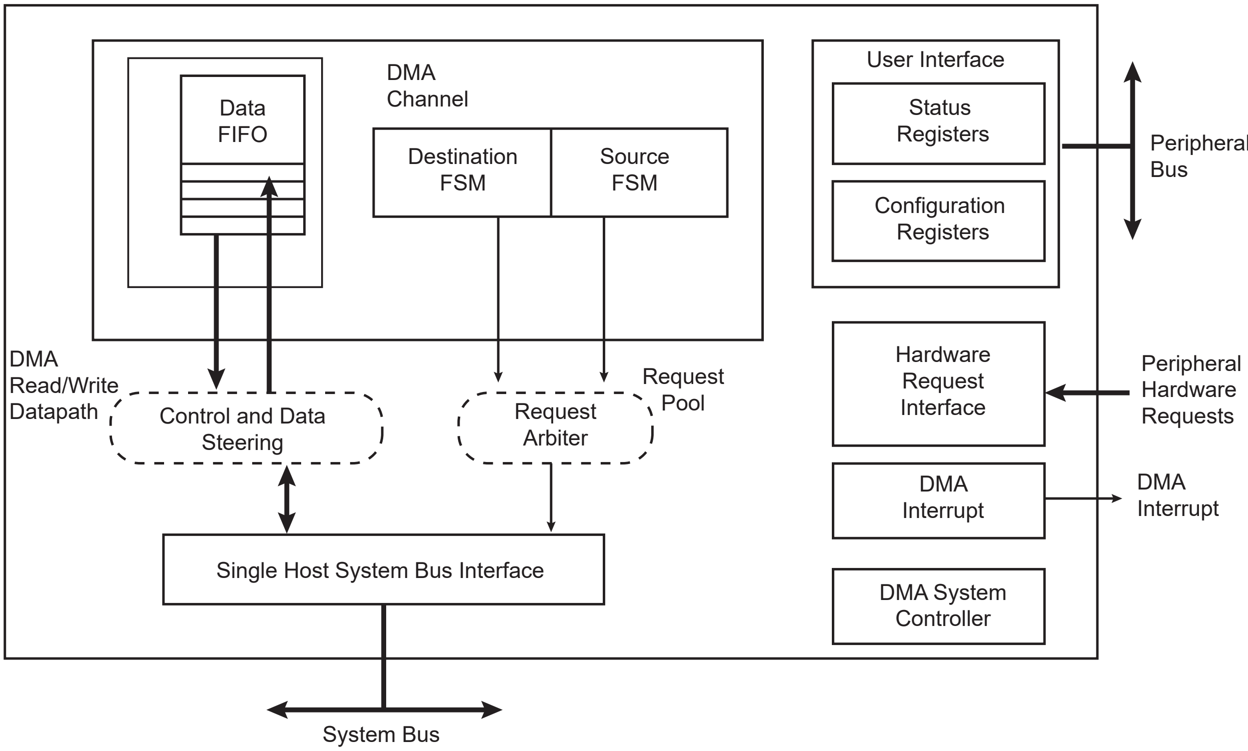
The DMA Controller (XDMAC) is a AXI-protocol central direct memory access controller. It performs peripheral data transfer and memory move operations over one or two bus ports through the unidirectional communication channel. Each channel is fully programmable and provides both peripheral or memory-to-memory transfers. The channel features are configurable at implementation.

14.2 Embedded Characteristics

- 1 System Bus Host Interface
- 32, 32 and 8 DMA Channels, Respectively, for XDMA0, XDMA1 and XDMA2
- 61 Hardware Requests
- 8448, 8448 and 2112 Kbytes Embedded FIFO, Respectively, for XDMA0, XDMA1 and XDMA2
- Supports Peripheral-to-Memory, Memory-to-Peripheral, or Memory-to-Memory Transfer Operations
- Peripheral DMA Operation Runs on Bytes (8-bit), Half-Word (16-bit) and Word (32-bit)
- Memory DMA Operation Runs on Bytes (8 bit), Half-Word (16-bit), Word (32-bit) and Double-Word (64-bit)
- Supports Hardware and Software Initiated Transfers
- Supports Linked List Operations
- Supports Incrementing or Fixed Addressing Mode
- Supports Programmable Independent Data Striding for Source and Destination
- Supports Programmable Independent Microblock Striding for Source and Destination
- Configurable Priority Group and Arbitration Policy
- Programmable AXI Burst Length
- Configuration Interface on Peripheral Bus
- XDMAC Architecture Includes Multiport FIFO
- Supports Multiple View Channel Descriptor
- Automatic Flush of Channel Trailing Bytes
- Automatic Coarse-Grain and Fine-Grain Clock Gating
- Hardware Acceleration of Memset Pattern
- Supports Configurable Quality of Service per Channel

14.3 Block Diagram

Figure 14-1. XDMAC Block Diagram



14.4 DMA Controller Peripheral Connections

DMA hardware requests are the same for XDMAC0 and XDMAC1. XDMAC2 is memory-to-memory dedicated, there is no hardware interface.

Table 14-1. DMA Channels Definitions (XDMAC0 and XDMAC1)

Instance Name	Channel Transmit/Receive	DMA Channel Hardware Interface Number	Comments
ADC	Receive	0	–
AES	Transmit	1	–
AES	Receive	2	–
FLEXCOM0	Receive	5	–
FLEXCOM0	Transmit	6	–
FLEXCOM1	Receive	7	–
FLEXCOM1	Transmit	8	–
FLEXCOM2	Receive	9	–
FLEXCOM2	Transmit	10	–
FLEXCOM3	Receive	11	–
FLEXCOM3	Transmit	12	–

...........continued

Instance Name	Channel Transmit/Receive	DMA Channel Hardware Interface Number	Comments
FLEXCOM4	Receive	13	–
FLEXCOM4	Transmit	14	–
FLEXCOM5	Receive	15	–
FLEXCOM5	Transmit	16	–
FLEXCOM6	Receive	17	–
FLEXCOM6	Transmit	18	–
FLEXCOM7	Receive	19	–
FLEXCOM7	Transmit	20	–
FLEXCOM8	Receive	21	–
FLEXCOM8	Transmit	22	–
FLEXCOM9	Receive	23	–
FLEXCOM9	Transmit	24	–
FLEXCOM10	Receive	25	–
FLEXCOM10	Transmit	26	–
FLEXCOM11	Receive	27	–
FLEXCOM11	Transmit	28	–
I2SMCC0	Receive	33	–
I2SMCC0	Transmit	34	–
I2SMCC1	Receive	35	–
I2SMCC1	Transmit	36	–
PDMC0	Receive	37	–
PDMC1	Receive	38	–
PWM	Transmit	39	–
QSPI0	Receive	40	–
QSPI0	Transmit	41	–
QSPI1	Receive	42	–
QSPI1	Transmit	43	–
SSC0	Receive	44	–
SSC0	Transmit	45	–
SSC1	Receive	46	–
SSC1	Transmit	47	–
SHA	Transmit	48	–
SPDIFRX	Receive	49	–
SPDIFTX	Transmit	50	–

...........continued			
Instance Name	**Channel Transmit/Receive**	**DMA Channel Hardware Interface Number**	**Comments**
TC0	Receive	51	–
TC1	Receive	52	–
TDES	Receive	53	–
TDES	Transmit	54	–
ASRC	R0	55	–
ASRC	T0	56	–
ASRC	R1	57	–
ASRC	T1	58	–
ASRC	R2	59	–
ASRC	T2	60	–
ASRC	R3	61	–
ASRC	T3	62	–
TC0_CHANNEL1	CPA	63	–
TC1_CHANNEL1	CPA	64	–
TC0_CHANNEL1	CPB	65	–
TC1_CHANNEL1	CPB	66	–
TC0_CHANNEL1	CPC	67	–
TC1_CHANNEL1	CPC	68	–
TC0_CHANNEL1	ETRG	69	–
TC1_CHANNEL1	ETRG	70	–
Reserved	–	71-126	Do not use
Reserved	–	127	Memory-to-memory transfer

14.5 Functional Description

14.5.1 Basic Definitions

Source Peripheral: Client device, memory mapped on the interconnection network, from where the XDMAC reads data. The source peripheral teams up with a destination peripheral to form a channel. A data read operation is scheduled when the peripheral transfer request is asserted.

Destination Peripheral: Client device, memory mapped on the interconnection network, to which the XDMAC writes. A write data operation is scheduled when the peripheral transfer request is asserted.

Channel: The data movement between source and destination creates a logical channel.

Stride: Number of address locations between successive elements/data measured in bytes.

Transfer Type: The transfer is hardware-synchronized when it is paced by the peripheral hardware request, otherwise the transfer is self-triggered (memory to memory transfer).

XDMAC Host Transfer: The host transfer is composed of a linked list of blocks. The channel address, control and configuration registers can be modified at the inter block boundary. The descriptor structure modifies the channel registers conditionally. Interrupts can be generated on a per block basis or when the end of linked list event occurs.

XDMAC Block: An XDMAC block is composed of a programmable number of microblocks. The channel configuration registers remain unchanged at the inter microblock boundary. The source and destination addresses are conditionally updated with a programmable signed number.

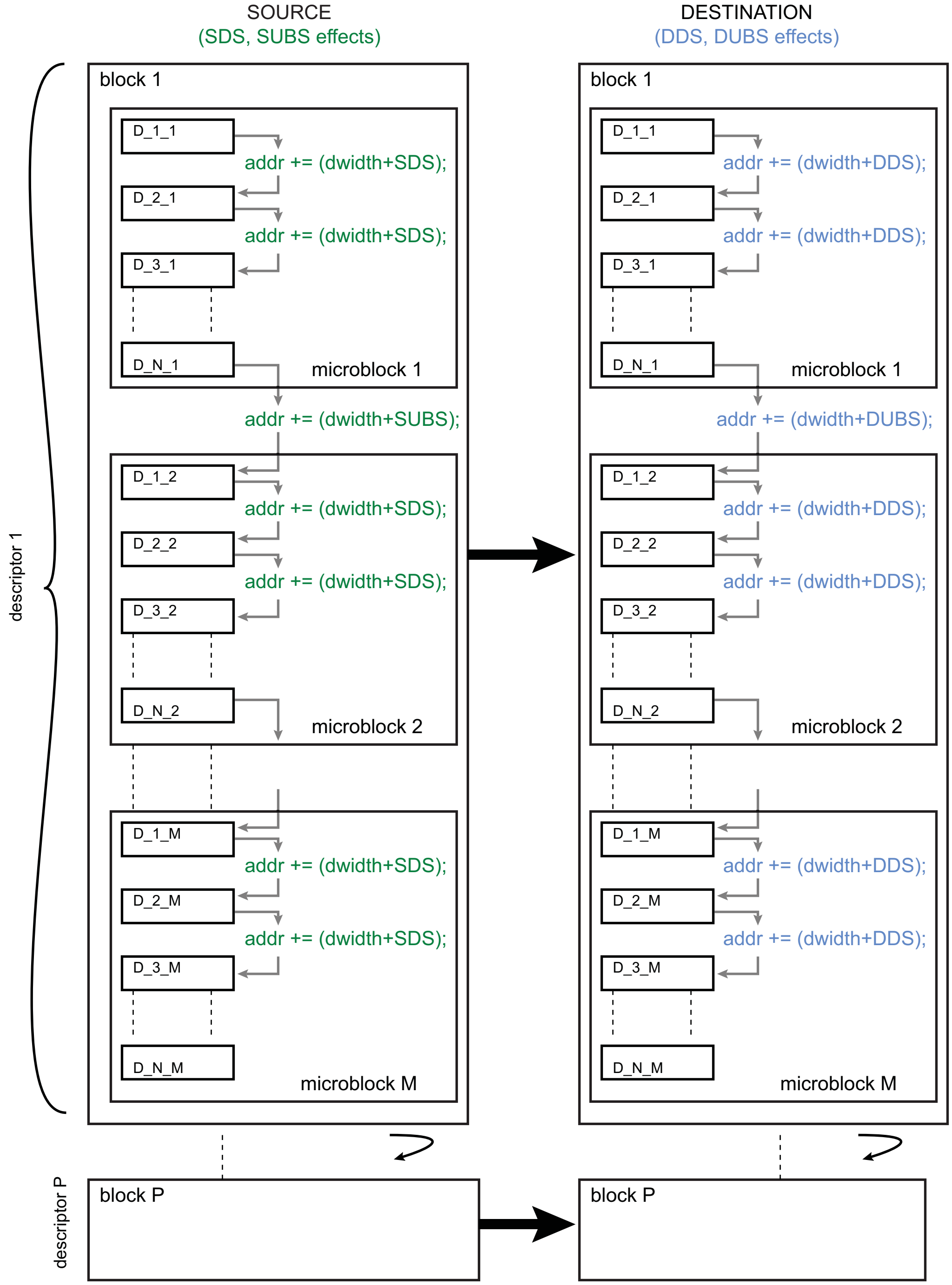
XDMAC Microblock: The microblock is composed of a programmable number of data. The channel configuration registers remain unchanged at the data boundary. The data address may be fixed (a FIFO location, a peripheral transmit or receive register), incrementing (a memory-mapped area) by a programmable signed number.

XDMAC Burst and Incomplete Burst: In order to improve the overall performance when accessing dynamic external memory, burst access is mandatory. Each data of the microblock is considered as a part of a memory burst. The programmable burst value indicates the largest memory burst allowed on a per channel basis. When the microblock length is not an integral multiple of the burst size, an incomplete burst is performed to read or write the last trailing bytes.

XDMAC Chunk and Incomplete Chunk: When a peripheral synchronized transfer is activated, the microblock splits into a number of data chunks. The chunk size is programmable. The larger the chunk is, the better the performance is. When the transfer size is not a multiple of the chunk size, the last chunk may be incomplete.

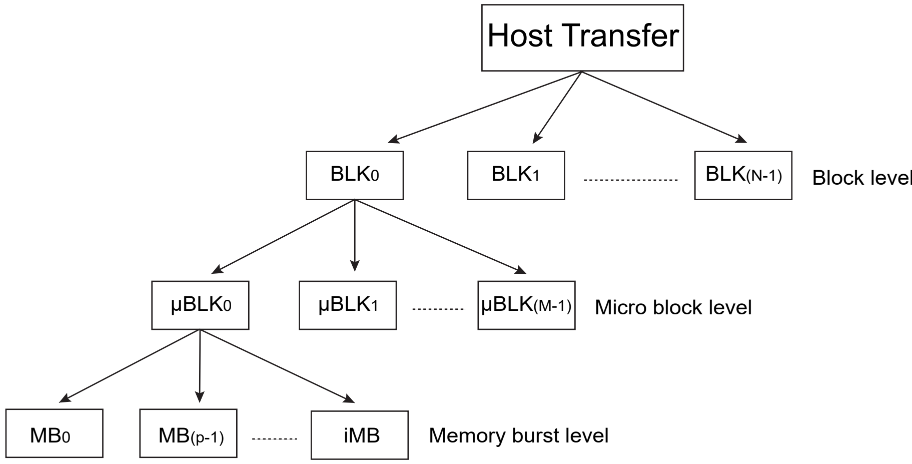
14.5.2 Data Striding Diagram

Figure 14-2. Data Striding Diagram



14.5.3 Transfer Hierarchy Diagrams

Figure 14-3. XDMAC Memory Transfer Hierarchy



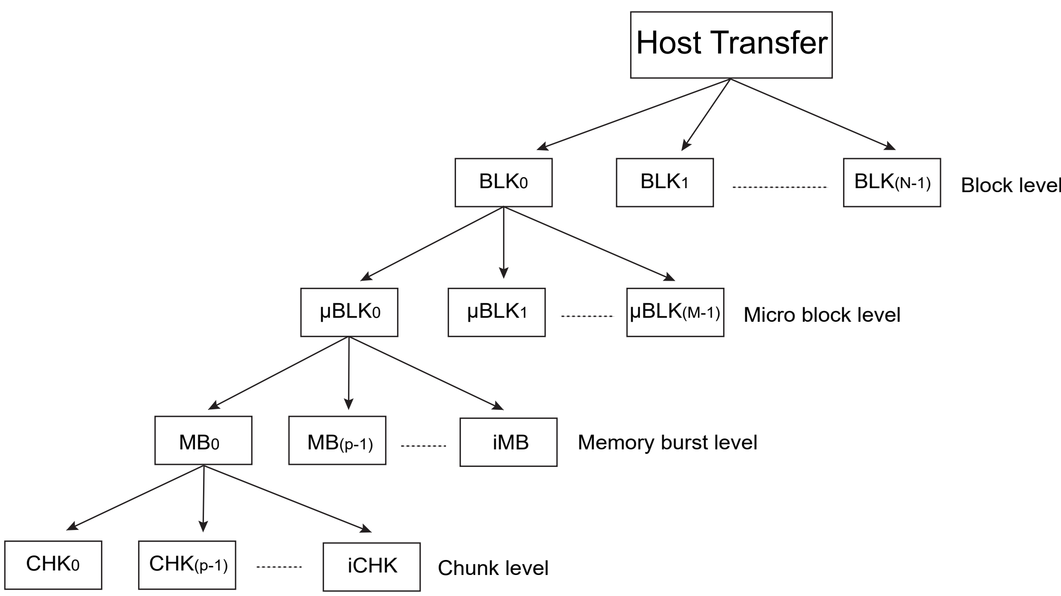
14.5.4 Peripheral Synchronized Transfer

A peripheral hardware request interface is used to control the pace of the chunk transfer. When a peripheral is ready to transmit or receive a chunk of data, it asserts its request line and the DMA Controller transfers a data to or from the memory to the peripheral.

14.5.4.1 Peripheral to Memory Transfer

XDMAC reads data from the source peripheral and writes to the destination memory location.

Figure 14-4. Peripheral to Memory Transfer Hierarchy



It is a peripheral synchronized transfer, which means the memory transaction is synchronized with the hardware trigger that comes from the corresponding peripheral. It is also possible to use software trigger to initiate data transfer. Peripheral to memory transfer has totally five levels of data transactions. They are Host, Block, Microblock, Burst, and Chunk level transactions. Host, Block, Microblock, and Burst level transactions work exactly the same way as explained earlier in the memory to memory data transfer section. In peripheral to memory data transfer, the burst level transaction is further split into chunk level data transaction to have higher granularity.

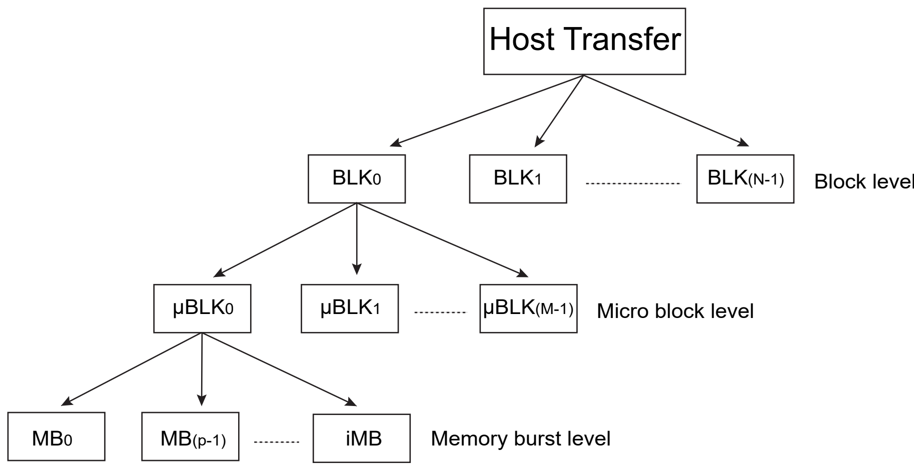
XDMAC Chunk and Incomplete Chunk: When a peripheral to memory transfer is activated, the burst level transaction is further split into a number of data chunks. The chunk size is configured in CSIZE field of XDMAC Channel Configuration Register (XDMAC_CCx). The chunk size denotes the number of 'data' to be transferred from the corresponding peripheral receive register to memory. In general, the chunk size is set as ‘1 data’ in most of the peripherals (example: - UART, SPI, TWI, etc.), as the maximum size of their receive register is ‘1 data’. In specific scenarios, the chunk size is chosen more than 1 data. For example, the data receive/input registers of AES and HSMCI modules can hold more than ‘1 data’. So, the chunk size can be chosen as '2/4/8/16 data' accordingly. In this case, the larger the chunk size is, the better the performance is. When the amount of data chunks read becomes equal to the memory burst size, the actual data transaction starts (as a memory burst). During ‘peripheral to memory’ transfer, the data chunks are first read and stored into XDMAC's internal FIFO buffer. If their size becomes equal to the memory burst size, the FIFO buffer gets flushed out automatically, which makes ‘memory burst transfer’. When the microblock size is not a multiple of the chunk size, the last chunk being transferred contains the last trailing data.

Note: If the chunk size is chosen as more than '1 data' for peripherals like UART, SPI, TWI, etc., XDMAC reads the same data register (receive/input register) multiple times. As a result, multiple copies of the same data are stored in the memory.

14.5.4.2 Memory to Peripheral Transfer

XDMAC reads data from source memory location and writes to the destination peripheral.

Figure 14-5. Memory to Peripheral Transfer Hierarchy



Memory to Peripheral transfer is also a peripheral synchronized transfer. It has totally four levels of data transactions. They are Host, Block, Microblock, and Chunk level transactions. Host, Block, and Microblock level transactions work exactly the same way as explained earlier in the memory to memory data transfer section. In memory to peripheral data transfer, the burst level transaction is not present. The microblock is directly split into chunk level data transaction.

XDMAC Chunk and Incomplete Chunk: When a memory to peripheral transfer is activated, the microblock level transaction is directly split into a number of data chunks. The chunk size is configured in CSIZE field of XDMAC Channel Configuration Register (XDMAC_CCx). The chunk size denotes the number of ‘data’ to be transferred from memory to the corresponding peripheral transmit register. In general, the chunk size is set as ‘1 data’ in most of the peripherals (example: - UART, SPI, TWI, etc.), as the maximum size of their transmit register is ‘1 data’. In specific scenarios, the chunk size is chosen more than 1 data. For example, the data transmit/output registers of AES and HSMCI modules can hold more than ‘1 data’. So, the chunk size can be chosen as '2/4/8/16 data' accordingly. In this case, the larger the chunk size is, the better the performance is. During ‘memory to peripheral’ transfer, the data chunks are immediately transferred when there is a hardware/software trigger. Memory burst size doesn't play any role here. When the microblock size is not a multiple of the chunk size, the last chunk being transferred contains the last trailing data.

Note: In case if the chunk size is chosen as more than '1 data' for peripherals like UART, SPI, TWI, etc., then XDMAC will overwrite the same data register (transmit/output register) with multiple data. As a result, only the last data gets transmitted.

14.5.4.3 Software Triggered Synchronized Transfer

The Peripheral hardware request can be software controlled using the SWREQ field of the XDMAC Global Channel Software Request Register (XDMAC_GSWR). The peripheral synchronized transfer is paced using a processor write access in the XDMAC_GSWR. Each bit of that register triggers a transfer request. The XDMAC Global Channel

Software Request Status Register (XDMAC_GSWS) indicates the status of the request; when set, the request is still pending.

14.5.5 XDMAC Transfer Software Operation

Note: When performing a memory-to-memory transfer, configure the field XDMAC_CCx.PERID (where 'x' is the index of the channel used for the transfer) to 0x7F.

14.5.5.1 Single Block Transfer With Single Microblock

1. Read the XDMAC Global Channel Status Register (XDMAC_GS) to select a free channel.
2. Clear the pending Interrupt Status bit(s) by reading the selected XDMAC Channel x Interrupt Status Register (XDMAC_CISx).
3. Write the XDMAC Channel x Source Address Register (XDMAC_CSAx) for channel x.
4. Write the XDMAC Channel x Destination Address Register (XDMAC_CDAx) for channel x.
5. Program field UBLEN in the XDMAC Channel x Microblock Control Register (XDMAC_CUBCx) with the number of data.
6. Program the XDMAC Channel x Configuration Register (XDMAC_CCx):
   a. Clear XDMAC_CCx.TYPE for a memory-to-memory transfer, otherwise set this bit.
   b. Configure XDMAC_CCx.MBSIZE to the memory burst size used.
   c. Configure XDMAC_CCx.SAM and DAM to Memory Addressing mode.
   d. Configure XDMAC_CCx.DSYNC to select the peripheral transfer direction.
   e. Set XDMAC_CCx.PROT to activate a secure channel.
   f. Configure XDMAC_CCx.CSIZE to configure the channel chunk size (only relevant for peripheral synchronized transfer).
   g. Configure XDMAC_CCx.DWIDTH to configure the transfer data width.
   h. Configure XDMAC_CCx.PERID to select the active hardware request line (only relevant for a peripheral synchronized transfer).
   i. Set XDMAC_CCx.SWREQ to use a software request (only relevant for a peripheral synchronized transfer).
7. Clear the following five registers:
   – XDMAC Channel x Next Descriptor Control Register (XDMAC_CNDCx)
   – XDMAC Channel x Block Control Register (XDMAC_CBCx)
   – XDMAC Channel x Data Stride Memory Set Pattern Register (XDMAC_CDS_MSPx)
   – XDMAC Channel x Source Microblock Stride Register (XDMAC_CSUSx)
   – XDMAC Channel x Destination Microblock Stride Register (XDMAC_CDUSx)
   This indicates that the linked list is disabled, there is only one block and striding is disabled.
8. Enable the Microblock interrupt by writing a '1' to bit BIE in the XDMAC Channel x Interrupt Enable Register (XDMAC_CIEx). Enable the Channel x Interrupt Enable bit by writing a '1' to bit IEx in the XDMAC Global Interrupt Enable Register (XDMAC_GIE).
9. Enable channel x by writing a '1' to bit ENx in the XDMAC Global Channel Enable Register (XDMAC_GE). XDMAC_GS.STx (XDMAC Channel x Status bit) is set by hardware.
10. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

14.5.5.2 Single Block Transfer With Multiple Microblock

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Write the XDMAC_CSAx register for channel x.
4. Write the XDMAC_CDAx register for channel x.
5. Program XDMAC_CUBCx.UBLEN with the number of data.
6. Program XDMAC_CCx register (see "Single Block Transfer With Single Microblock").
7. Program XDMAC_CBCx.BLEN with the number of microblocks of data.
8. Clear the following registers:

   – XDMAC_CNDCx
   – XDMAC_CDS_MSPx
   – XDMAC_CSUSx XDMAC_CDUSx
     This indicates that the linked list is disabled and striding is disabled.
9. Enable the Block interrupt by writing a ‘1’ to XDMAC_CIEx.BIE, enable the Channel x Interrupt Enable bit by writing a ‘1’ to XDMAC_GIEx.IEx.
10. Enable channel x by writing a ‘1’ to the XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
11. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

14.5.5.3 Host Transfer

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Build a linked list of transfer descriptors in memory. The descriptor view is programmable on a per descriptor basis. The linked list items structure must be word aligned. MBR_UBC.NDE must be configured to 0 in the last descriptor to terminate the list.
4. Configure field NDA in the XDMAC Channel x Next Descriptor Address Register (XDMAC_CNDAx) with the first descriptor address .
5. Configure the XDMAC_CNDCx register:
   a. Set XDMAC_CNDCx.NDE to enable the descriptor fetch.
   b. Set XDMAC_CNDCx.NDSUP to update the source address at the descriptor fetch time, otherwise clear this bit.
   c. Set XDMAC_CNDCx.NDDUP to update the destination address at the descriptor fetch time, otherwise clear this bit.
   d. Configure XDMAC_CNDCx.NDVIEW to define the length of the first descriptor.
6. Enable the End of Linked List interrupt by writing a ‘1’ to XDMAC_CIEx.LIE.
7. Enable channel x by writing a ‘1’ to XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
8. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

14.5.5.4 Disabling A Channel Before Transfer Completion

Under normal operation, the software enables a channel by writing a ‘1’ to XDMAC_GE.ENx, then the hardware disables a channel on transfer completion by clearing bit XDMAC_GS.STx. To disable a channel, write a ‘1’ to bit XDMAC_GD.DIx and poll the XDMAC_GS register.

14.6 Linked List Descriptor Operation

14.6.1 Linked List Descriptor View

14.6.1.1 Channel Next Descriptor View 0–3 Structures

Table 14-2. Channel Next Descriptor View 0–3 Structures

Channel Next Descriptor	Offset	Structure member	Name
View 0 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Transfer Address Member	MBR_TA

...........continued			
Channel Next Descriptor	**Offset**	**Structure member**	**Name**
View 1 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
View 2 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
	DSCR_ADDR+0x10	Configuration Register	MBR_CFG
View 3 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
	DSCR_ADDR+0x10	Configuration Member	MBR_CFG
	DSCR_ADDR+0x14	Block Control Member	MBR_BC
	DSCR_ADDR+0x18	Data Stride Member	MBR_DS
	DSCR_ADDR+0x1C	Source Microblock Stride Member	MBR_SUS
	DSCR_ADDR+0x20	Destination Microblock Stride Member	MBR_DUS

14.6.2 Descriptor Structure Members Description

14.6.2.1 Descriptor Structure Microblock Control Member

Name: MBR_UBC
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		QOS[1:0]		NVIEW[1:0]		NDEN	NSEN	NDE
Access		R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	UBLEN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	UBLEN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	UBLEN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bits 30:29 – QOS[1:0] Channel Quality of Service Level
This field indicates the current quality of service level for the channel. Refer to the section “MATRIX”.

Bits 28:27 – NVIEW[1:0] Next Descriptor View

Value	Name	Description
0	NDV0	Next Descriptor View 0
1	NDV1	Next Descriptor View 1
2	NDV2	Next Descriptor View 2
3	NDV3	Next Descriptor View 3

Bit 26 – NDEN Next Descriptor Destination Update

Value	Description
0	Destination parameters remain unchanged.
1	Destination parameters are updated when the descriptor is retrieved.

Bit 25 – NSEN Next Descriptor Source Update

Value	Description
0	Source parameters remain unchanged.
1	Source parameters are updated when the descriptor is retrieved.

Bit 24 – NDE Next Descriptor Enable

Value	Description
0	Descriptor fetch is disabled.
1	Descriptor fetch is enabled.

Bits 23:0 – UBLEN[23:0] Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

14.7 XDMAC Maintenance Software Operations

14.7.1 Disabling a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

14.7.2 Suspending a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

14.7.3 Flushing a Channel

A FIFO flush command is issued by writing to the XDMAC_SWF register. The content of the FIFO is written to memory. XDMAC_CISx.FIS (End of Flush Interrupt Status bit) is set when the last byte is successfully transferred to memory. The channel is not disabled. The flush operation is not blocking, meaning that read operation can be scheduled during the flush write operation. The flush operation is only relevant for peripheral to memory transfer where pending peripheral bytes are buffered into the channel FIFO.

14.7.4 Maintenance Operation Priority

14.7.4.1 Disable Operation Priority

- When a disable request occurs on a suspended channel, the XDMAC_GWS.WSx (Channel x Write Suspend bit) is cleared. If the transfer is source peripheral synchronized, the pending bytes are drained to memory. The bit XDMAC_CISx.DIS is set.
- When a disable request follows a flush request, if the flush last transaction is not yet scheduled, the flush request is discarded and the disable procedure is applied. Bit XDMAC_CISx.FIS is not set. Bit XDMAC_CISx.DIS is set when the disable request is completed. If the flush request transaction is already scheduled, the XDMAC_CISx.FIS is set. XDMAC_CISx.DIS is also set when the disable request is completed.

14.7.4.2 Flush Operation Priority

- When a flush request occurs on a suspended channel, if there are pending bytes in the FIFO, they are written out to memory, XDMAC_CISx.FIS is set. If the FIFO is empty, XDMAC_CISx.FIS is also set.
- If the flush operation is performed after a disable request, the flush command is ignored. XDMAC_CISx.FIS is not set.

14.7.4.3 Suspend Operation Priority

If the suspend operation is performed after a disable request, the write suspend operation is ignored.

14.8 XDMAC Software Requirements

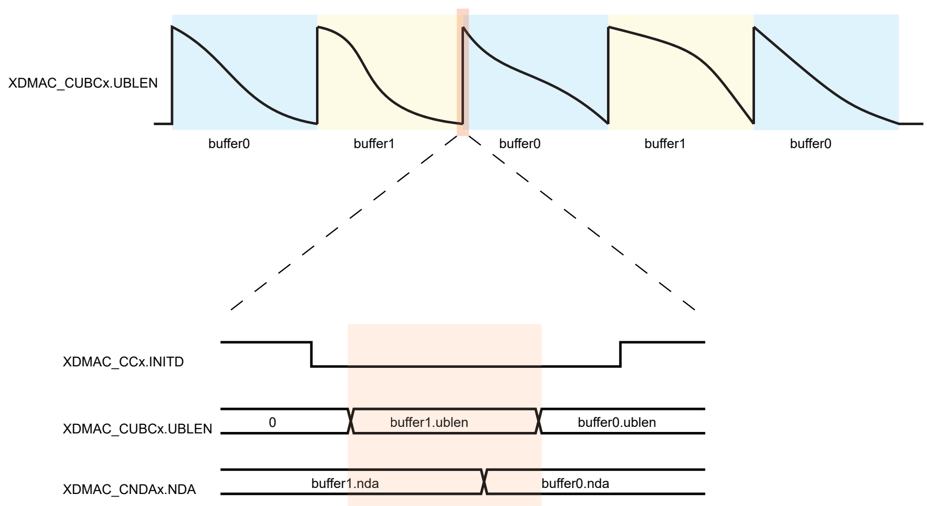
- Write operations to channel registers are not be performed in an active channel after the channel is enabled. If any channel parameters must be reprogrammed, this can only be done after disabling the XDMAC channel.
- XDMAC_CSAx and XDMAC_CDAx channel registers must be programmed with a byte, half-word, word or double-word aligned address depending on the Channel x Data Width field (DWIDTH) of the XDMAC Channel x Configuration Register. When a memory-to-peripheral transfer is performed, the XDMAC_CSAx address register has no alignment requirement.
- When a memory-to-memory transfer is performed, configure the field XDMAC_CCx.PERID (where ‘x’ is the index of the channel used for the transfer) to peripheral ID 127 (refer to the table “Peripheral Identifiers”).

- When XDMAC_CC.INITD is set to 0, XDMAC_CUBC.UBLEN and XDMAC_CNDA.NDA field values are unreliable when the descriptor is being updated. The following procedure applies to get the buffer descriptor identifier and the residual bytes:

```
Read XDMAC_CNDAx.NDA(nda0)
Read XDMAC_CCx.INITD(initd0)
Read XDMAC_CCx.INITD(initd0)
Read XDMAC_CUBCx.UBLEN(ublen)
Read XDMAC_CCx.INITD(initd1)
Read XDMA_CNDAx.NDA(nda1)
If (nda0 == nda1 && initd0 == 1 && initd1 == 1).
Then the ublen is correct, the buffer id is nda.
Else retry
```

  See the figure below.
- Each DMA channel can be configured in either Secure or Non-secure mode independently. When a DMA channel is secure, its related global register fields and its channel registers cannot be modified nor read by non-secure software. Such non-secure reads return zero.

Figure 14-6. INITD Timing Diagram



14.9 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	XDMAC_GTYPE	31:24								
		23:16		NB_REQ[6:0]						
		15:8	FIFO_SZ[10:3]							
		7:0	FIFO_SZ[2:0]			NB_CH[4:0]				
0x04	XDMAC_GCFG	31:24	RDSG[3:0]				RDLP[3:0]			
		23:16	RDMP[3:0]				RDHP[3:0]			
		15:8	WRLP[3:0]				WRMP[3:0]			
		7:0	WRHP[3:0]				CGDISIF	CGDISFIFO	CGDISPIPE	CGDISREG
0x08	XDMAC_GWAC	31:24								
		23:16								
		15:8	PW3[3:0]				PW2[3:0]			
		7:0	PW1[3:0]				PW0[3:0]			
0x0C	XDMAC_GIE	31:24	IE31	IE30	IE29	IE28	IE27	IE26	IE25	IE24
		23:16	IE23	IE22	IE21	IE20	IE19	IE18	IE17	IE16
		15:8	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8
		7:0	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
0x10	XDMAC_GID	31:24	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24
		23:16	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
		15:8	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
		7:0	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x14	XDMAC_GIM	31:24	IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24
		23:16	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
		15:8	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
		7:0	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
0x18	XDMAC_GIS	31:24	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24
		23:16	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
		15:8	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
		7:0	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0x1C	XDMAC_GE	31:24	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24
		23:16	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
		15:8	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
		7:0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
0x20	XDMAC_GD	31:24	DI31	DI30	DI29	DI28	DI27	DI26	DI25	DI24
		23:16	DI23	DI22	DI21	DI20	DI19	DI18	DI17	DI16
		15:8	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
		7:0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0x24	XDMAC_GS	31:24	ST31	ST30	ST29	ST28	ST27	ST26	ST25	ST24
		23:16	ST23	ST22	ST21	ST20	ST19	ST18	ST17	ST16
		15:8	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
		7:0	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
0x28	XDMAC_GRSS	31:24	RSS31	RSS30	RSS29	RSS28	RSS27	RSS26	RSS25	RSS24
		23:16	RSS23	RSS22	RSS21	RSS20	RSS19	RSS18	RSS17	RSS16
		15:8	RSS15	RSS14	RSS13	RSS12	RSS11	RSS10	RSS9	RSS8
		7:0	RSS7	RSS6	RSS5	RSS4	RSS3	RSS2	RSS1	RSS0
0x2C	XDMAC_GWSS	31:24	WSS31	WSS30	WSS29	WSS28	WSS27	WSS26	WSS25	WSS24
		23:16	WSS23	WSS22	WSS21	WSS20	WSS19	WSS18	WSS17	WSS16
		15:8	WSS15	WSS14	WSS13	WSS12	WSS11	WSS10	WSS9	WSS8
		7:0	WSS7	WSS6	WSS5	WSS4	WSS3	WSS2	WSS1	WSS0
0x30	XDMAC_GRS	31:24	RS31	RS30	RS29	RS28	RS27	RS26	RS25	RS24
		23:16	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16
		15:8	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
		7:0	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
0x34	XDMAC_GRR	31:24	RR31	RR30	RR29	RR28	RR27	RR26	RR25	RR24
		23:16	RR23	RR22	RR21	RR20	RR19	RR18	RR17	RR16
		15:8	RR15	RR14	RR13	RR12	RR11	RR10	RR9	RR8
		7:0	RR7	RR6	RR5	RR4	RR3	RR2	RR1	RR0

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x38	XDMAC_GWS	31:24	WS31	WS30	WS29	WS28	WS27	WS26	WS25	WS24
		23:16	WS23	WS22	WS21	WS20	WS19	WS18	WS17	WS16
		15:8	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
		7:0	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
0x3C	XDMAC_GWR	31:24	WR31	WR30	WR29	WR28	WR27	WR26	WR25	WR24
		23:16	WR23	WR22	WR21	WR20	WR19	WR18	WR17	WR16
		15:8	WR15	WR14	WR13	WR12	WR11	WR10	WR9	WR8
		7:0	WR7	WR6	WR5	WR4	WR3	WR2	WR1	WR0
0x40	XDMAC_GRWS	31:24	RWS31	RWS30	RWS29	RWS28	RWS27	RWS26	RWS25	RWS24
		23:16	RWS23	RWS22	RWS21	RWS20	RWS19	RWS18	RWS17	RWS16
		15:8	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8
		7:0	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
0x44	XDMAC_GRWR	31:24	RWR31	RWR30	RWR29	RWR28	RWR27	RWR26	RWR25	RWR24
		23:16	RWR23	RWR22	RWR21	RWR20	RWR19	RWR18	RWR17	RWR16
		15:8	RWR15	RWR14	RWR13	RWR12	RWR11	RWR10	RWR9	RWR8
		7:0	RWR7	RWR6	RWR5	RWR4	RWR3	RWR2	RWR1	RWR0
0x48	XDMAC_GSWR	31:24	SWREQ31	SWREQ30	SWREQ29	SWREQ28	SWREQ27	SWREQ26	SWREQ25	SWREQ24
		23:16	SWREQ23	SWREQ22	SWREQ21	SWREQ20	SWREQ19	SWREQ18	SWREQ17	SWREQ16
		15:8	SWREQ15	SWREQ14	SWREQ13	SWREQ12	SWREQ11	SWREQ10	SWREQ9	SWREQ8
		7:0	SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0
0x4C	XDMAC_GSWS	31:24	SWRS31	SWRS30	SWRS29	SWRS28	SWRS27	SWRS26	SWRS25	SWRS24
		23:16	SWRS23	SWRS22	SWRS21	SWRS20	SWRS19	SWRS18	SWRS17	SWRS16
		15:8	SWRS15	SWRS14	SWRS13	SWRS12	SWRS11	SWRS10	SWRS9	SWRS8
		7:0	SWRS7	SWRS6	SWRS5	SWRS4	SWRS3	SWRS2	SWRS1	SWRS0
0x50	XDMAC_GSWF	31:24	SWF31	SWF30	SWF29	SWF28	SWF27	SWF26	SWF25	SWF24
		23:16	SWF23	SWF22	SWF21	SWF20	SWF19	SWF18	SWF17	SWF16
		15:8	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
		7:0	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
0x54 ... 0x5F	Reserved									
0x60	XDMAC_CIE0	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x64	XDMAC_CID0	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x68	XDMAC_CIM0	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x6C	XDMAC_CIS0	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x70	XDMAC_CSA0	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x74	XDMAC_CDA0	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x78	XDMAC_CNDA0	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x7C	XDMAC_CNDC0	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x80	XDMAC_CUBC0	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x84	XDMAC_CBC0	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x88	XDMAC_CC0	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x8C	XDMAC_CDS_MSP0	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x90	XDMAC_CSUS0	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x94	XDMAC_CDUS0	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x98	XDMAC_CTCS0	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x9C ... 0x9F	Reserved									
0xA0	XDMAC_CIE1	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0xA4	XDMAC_CID1	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0xA8	XDMAC_CIM1	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0xAC	XDMAC_CIS1	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xB0	XDMAC_CSA1	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0xB4	XDMAC_CDA1	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB8	XDMAC_CNDA1	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0xBC	XDMAC_CNDC1	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0xC0	XDMAC_CUBC1	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0xC4	XDMAC_CBC1	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0xC8	XDMAC_CC1	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0xCC	XDMAC_CDS_MSP1	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0xD0	XDMAC_CSUS1	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0xD4	XDMAC_CDUS1	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0xD8	XDMAC_CTCS1	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0xDC ... 0xDF	Reserved									
0xE0	XDMAC_CIE2	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0xE4	XDMAC_CID2	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0xE8	XDMAC_CIM2	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0xEC	XDMAC_CIS2	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xF0	XDMAC_CSA2	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xF4	XDMAC_CDA2	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0xF8	XDMAC_CNDA2	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0xFC	XDMAC_CNDC2	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0100	XDMAC_CUBC2	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0104	XDMAC_CBC2	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0108	XDMAC_CC2	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x010C	XDMAC_CDS_MSP2	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0110	XDMAC_CSUS2	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0114	XDMAC_CDUS2	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0118	XDMAC_CTCS2	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x011C ... 0x011F	Reserved									
0x0120	XDMAC_CIE3	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0124	XDMAC_CID3	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0128	XDMAC_CIM3	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x012C	XDMAC_CIS3	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0130	XDMAC_CSA3	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0134	XDMAC_CDA3	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0138	XDMAC_CNDA3	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x013C	XDMAC_CNDC3	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0140	XDMAC_CUBC3	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0144	XDMAC_CBC3	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0148	XDMAC_CC3	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x014C	XDMAC_CDS_MSP3	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0150	XDMAC_CSUS3	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0154	XDMAC_CDUS3	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0158	XDMAC_CTCS3	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x015C ... 0x015F	Reserved									
0x0160	XDMAC_CIE4	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0164	XDMAC_CID4	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0168	XDMAC_CIM4	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x016C	XDMAC_CIS4	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0170	XDMAC_CSA4	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0174	XDMAC_CDA4	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0178	XDMAC_CNDA4	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x017C	XDMAC_CNDC4	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0180	XDMAC_CUBC4	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0184	XDMAC_CBC4	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0188	XDMAC_CC4	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x018C	XDMAC_CDS_MSP4	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0190	XDMAC_CSUS4	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0194	XDMAC_CDUS4	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0198	XDMAC_CTCS4	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x019C ... 0x019F	Reserved									
0x01A0	XDMAC_CIE5	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x01A4	XDMAC_CID5	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01A8	XDMAC_CIM5	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x01AC	XDMAC_CIS5	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01B0	XDMAC_CSA5	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01B4	XDMAC_CDA5	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01B8	XDMAC_CNDA5	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x01BC	XDMAC_CNDC5	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x01C0	XDMAC_CUBC5	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x01C4	XDMAC_CBC5	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x01C8	XDMAC_CC5	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x01CC	XDMAC_CDS_MSP5	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x01D0	XDMAC_CSUS5	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x01D4	XDMAC_CDUS5	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x01D8	XDMAC_CTCS5	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x01DC ... 0x01DF	Reserved									
0x01E0	XDMAC_CIE6	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
...........continued										
0x01E4	XDMAC_CID6	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x01E8	XDMAC_CIM6	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x01EC	XDMAC_CIS6	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01F0	XDMAC_CSA6	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01F4	XDMAC_CDA6	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01F8	XDMAC_CNDA6	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x01FC	XDMAC_CNDC6	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0200	XDMAC_CUBC6	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0204	XDMAC_CBC6	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0208	XDMAC_CC6	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x020C	XDMAC_CDS_MSP6	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0210	XDMAC_CSUS6	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0214	XDMAC_CDUS6	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0218	XDMAC_CTCS6	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x021C ... 0x021F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0220	XDMAC_CIE7	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0224	XDMAC_CID7	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0228	XDMAC_CIM7	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x022C	XDMAC_CIS7	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0230	XDMAC_CSA7	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0234	XDMAC_CDA7	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0238	XDMAC_CNDA7	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x023C	XDMAC_CNDC7	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0240	XDMAC_CUBC7	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0244	XDMAC_CBC7	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0248	XDMAC_CC7	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x024C	XDMAC_CDS_MSP7	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0250	XDMAC_CSUS7	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0254	XDMAC_CDUS7	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0258	XDMAC_CTCS7	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x025C ... 0x025F	Reserved									
0x0260	XDMAC_CIE8	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0264	XDMAC_CID8	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0268	XDMAC_CIM8	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x026C	XDMAC_CIS8	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0270	XDMAC_CSA8	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0274	XDMAC_CDA8	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0278	XDMAC_CNDA8	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x027C	XDMAC_CNDC8	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0280	XDMAC_CUBC8	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0284	XDMAC_CBC8	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0288	XDMAC_CC8	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x028C	XDMAC_CDS_MSP8	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0290	XDMAC_CSUS8	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0294	XDMAC_CDUS8	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0298	XDMAC_CTCS8	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x029C ... 0x029F	Reserved									
0x02A0	XDMAC_CIE9	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x02A4	XDMAC_CID9	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x02A8	XDMAC_CIM9	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x02AC	XDMAC_CIS9	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02B0	XDMAC_CSA9	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02B4	XDMAC_CDA9	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02B8	XDMAC_CNDA9	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x02BC	XDMAC_CNDC9	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x02C0	XDMAC_CUBC9	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x02C4	XDMAC_CBC9	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x02C8	XDMAC_CC9	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x02CC	XDMAC_CDS_MSP9	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02D0	XDMAC_CSUS9	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x02D4	XDMAC_CDUS9	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x02D8	XDMAC_CTCS9	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x02DC ... 0x02DF	Reserved									
0x02E0	XDMAC_CIE10	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x02E4	XDMAC_CID10	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x02E8	XDMAC_CIM10	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x02EC	XDMAC_CIS10	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02F0	XDMAC_CSA10	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02F4	XDMAC_CDA10	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02F8	XDMAC_CNDA10	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x02FC	XDMAC_CNDC10	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0300	XDMAC_CUBC10	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0304	XDMAC_CBC10	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0308	XDMAC_CC10	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x030C	XDMAC_CDS_MSP10	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0310	XDMAC_CSUS10	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0314	XDMAC_CDUS10	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0318	XDMAC_CTCS10	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x031C ... 0x031F	Reserved									
0x0320	XDMAC_CIE11	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0324	XDMAC_CID11	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0328	XDMAC_CIM11	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x032C	XDMAC_CIS11	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0330	XDMAC_CSA11	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0334	XDMAC_CDA11	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0338	XDMAC_CNDA11	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x033C	XDMAC_CNDC11	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0340	XDMAC_CUBC11	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0344	XDMAC_CBC11	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0348	XDMAC_CC11	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x034C	XDMAC_CDS_MSP11	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0350	XDMAC_CSUS11	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0354	XDMAC_CDUS11	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0358	XDMAC_CTCS11	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x035C ... 0x035F	Reserved									
0x0360	XDMAC_CIE12	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0364	XDMAC_CID12	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0368	XDMAC_CIM12	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x036C	XDMAC_CIS12	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0370	XDMAC_CSA12	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0374	XDMAC_CDA12	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0378	XDMAC_CNDA12	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x037C	XDMAC_CNDC12	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0380	XDMAC_CUBC12	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0384	XDMAC_CBC12	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0388	XDMAC_CC12	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x038C	XDMAC_CDS_MSP12	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0390	XDMAC_CSUS12	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0394	XDMAC_CDUS12	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0398	XDMAC_CTCS12	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x039C ... 0x039F	Reserved									
0x03A0	XDMAC_CIE13	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x03A4	XDMAC_CID13	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x03A8	XDMAC_CIM13	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x03AC	XDMAC_CIS13	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03B0	XDMAC_CSA13	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03B4	XDMAC_CDA13	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03B8	XDMAC_CNDA13	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x03BC	XDMAC_CNDC13	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x03C0	XDMAC_CUBC13	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x03C4	XDMAC_CBC13	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x03C8	XDMAC_CC13	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x03CC	XDMAC_CDS_MSP13	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x03D0	XDMAC_CSUS13	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x03D4	XDMAC_CDUS13	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x03D8	XDMAC_CTCS13	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x03DC ... 0x03DF	Reserved									
0x03E0	XDMAC_CIE14	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x03E4	XDMAC_CID14	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x03E8	XDMAC_CIM14	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x03EC	XDMAC_CIS14	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03F0	XDMAC_CSA14	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03F4	XDMAC_CDA14	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03F8	XDMAC_CNDA14	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03FC	XDMAC_CNDC14	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0400	XDMAC_CUBC14	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0404	XDMAC_CBC14	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0408	XDMAC_CC14	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x040C	XDMAC_CDS_MSP14	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0410	XDMAC_CSUS14	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0414	XDMAC_CDUS14	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0418	XDMAC_CTCS14	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x041C ... 0x041F	Reserved									
0x0420	XDMAC_CIE15	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0424	XDMAC_CID15	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0428	XDMAC_CIM15	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x042C	XDMAC_CIS15	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0430	XDMAC_CSA15	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0434	XDMAC_CDA15	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0438	XDMAC_CNDA15	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x043C	XDMAC_CNDC15	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0440	XDMAC_CUBC15	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0444	XDMAC_CBC15	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0448	XDMAC_CC15	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x044C	XDMAC_CDS_MSP15	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0450	XDMAC_CSUS15	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0454	XDMAC_CDUS15	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0458	XDMAC_CTCS15	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x045C ... 0x045F	Reserved									
0x0460	XDMAC_CIE16	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0464	XDMAC_CID16	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0468	XDMAC_CIM16	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x046C	XDMAC_CIS16	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0470	XDMAC_CSA16	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0474	XDMAC_CDA16	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0478	XDMAC_CNDA16	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x047C	XDMAC_CNDC16	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0480	XDMAC_CUBC16	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0484	XDMAC_CBC16	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0488	XDMAC_CC16	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x048C	XDMAC_CDS_MSP16	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0490	XDMAC_CSUS16	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0494	XDMAC_CDUS16	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0498	XDMAC_CTCS16	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x049C ... 0x049F	Reserved									
0x04A0	XDMAC_CIE17	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x04A4	XDMAC_CID17	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x04A8	XDMAC_CIM17	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x04AC	XDMAC_CIS17	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04B0	XDMAC_CSA17	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x04B4	XDMAC_CDA17	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x04B8	XDMAC_CNDA17	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x04BC	XDMAC_CNDC17	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x04C0	XDMAC_CUBC17	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x04C4	XDMAC_CBC17	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x04C8	XDMAC_CC17	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x04CC	XDMAC_CDS_MSP17	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x04D0	XDMAC_CSUS17	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x04D4	XDMAC_CDUS17	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x04D8	XDMAC_CTCS17	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x04DC ... 0x04DF	Reserved									
0x04E0	XDMAC_CIE18	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x04E4	XDMAC_CID18	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x04E8	XDMAC_CIM18	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM

..........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04EC	XDMAC_CIS18	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x04F0	XDMAC_CSA18	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x04F4	XDMAC_CDA18	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x04F8	XDMAC_CNDA18	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x04FC	XDMAC_CNDC18	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0500	XDMAC_CUBC18	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0504	XDMAC_CBC18	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0508	XDMAC_CC18	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x050C	XDMAC_CDS_MSP18	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0510	XDMAC_CSUS18	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0514	XDMAC_CDUS18	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0518	XDMAC_CTCS18	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x051C ... 0x051F	Reserved									
0x0520	XDMAC_CIE19	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0524	XDMAC_CID19	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0528	XDMAC_CIM19	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x052C	XDMAC_CIS19	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0530	XDMAC_CSA19	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0534	XDMAC_CDA19	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0538	XDMAC_CNDA19	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x053C	XDMAC_CNDC19	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0540	XDMAC_CUBC19	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0544	XDMAC_CBC19	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0548	XDMAC_CC19	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x054C	XDMAC_CDS_MSP19	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0550	XDMAC_CSUS19	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0554	XDMAC_CDUS19	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0558	XDMAC_CTCS19	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x055C ... 0x055F	Reserved									
0x0560	XDMAC_CIE20	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0564	XDMAC_CID20	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0568	XDMAC_CIM20	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x056C	XDMAC_CIS20	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0570	XDMAC_CSA20	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0574	XDMAC_CDA20	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0578	XDMAC_CNDA20	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x057C	XDMAC_CNDC20	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0580	XDMAC_CUBC20	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0584	XDMAC_CBC20	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0588	XDMAC_CC20	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x058C	XDMAC_CDS_MSP20	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0590	XDMAC_CSUS20	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0594	XDMAC_CDUS20	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0598	XDMAC_CTCS20	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x059C ... 0x059F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x05A0	XDMAC_CIE21	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x05A4	XDMAC_CID21	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x05A8	XDMAC_CIM21	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x05AC	XDMAC_CIS21	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x05B0	XDMAC_CSA21	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x05B4	XDMAC_CDA21	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x05B8	XDMAC_CNDA21	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x05BC	XDMAC_CNDC21	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x05C0	XDMAC_CUBC21	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x05C4	XDMAC_CBC21	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x05C8	XDMAC_CC21	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x05CC	XDMAC_CDS_MSP21	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x05D0	XDMAC_CSUS21	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x05D4	XDMAC_CDUS21	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x05D8	XDMAC_CTCS21	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x05DC ... 0x05DF	Reserved									
0x05E0	XDMAC_CIE22	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x05E4	XDMAC_CID22	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x05E8	XDMAC_CIM22	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x05EC	XDMAC_CIS22	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x05F0	XDMAC_CSA22	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x05F4	XDMAC_CDA22	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x05F8	XDMAC_CNDA22	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x05FC	XDMAC_CNDC22	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0600	XDMAC_CUBC22	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0604	XDMAC_CBC22	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0608	XDMAC_CC22	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x060C	XDMAC_CDS_MSP22	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0610	XDMAC_CSUS22	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0614	XDMAC_CDUS22	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0618	XDMAC_CTCS22	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x061C ... 0x061F	Reserved									
0x0620	XDMAC_CIE23	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0624	XDMAC_CID23	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0628	XDMAC_CIM23	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x062C	XDMAC_CIS23	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0630	XDMAC_CSA23	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0634	XDMAC_CDA23	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0638	XDMAC_CNDA23	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x063C	XDMAC_CNDC23	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0640	XDMAC_CUBC23	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0644	XDMAC_CBC23	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0648	XDMAC_CC23	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x064C	XDMAC_CDS_MSP 23	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0650	XDMAC_CSUS23	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0654	XDMAC_CDUS23	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0658	XDMAC_CTCS23	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x065C ... 0x065F	Reserved									
0x0660	XDMAC_CIE24	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0664	XDMAC_CID24	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0668	XDMAC_CIM24	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x066C	XDMAC_CIS24	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0670	XDMAC_CSA24	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0674	XDMAC_CDA24	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0678	XDMAC_CNDA24	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x067C	XDMAC_CNDC24	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0680	XDMAC_CUBC24	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0684	XDMAC_CBC24	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0688	XDMAC_CC24	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x068C	XDMAC_CDS_MSP24	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0690	XDMAC_CSUS24	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0694	XDMAC_CDUS24	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0698	XDMAC_CTCS24	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x069C ... 0x069F	Reserved									
0x06A0	XDMAC_CIE25	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x06A4	XDMAC_CID25	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x06A8	XDMAC_CIM25	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x06AC	XDMAC_CIS25	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x06B0	XDMAC_CSA25	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x06B4	XDMAC_CDA25	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x06B8	XDMAC_CNDA25	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x06BC	XDMAC_CNDC25	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x06C0	XDMAC_CUBC25	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x06C4	XDMAC_CBC25	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x06C8	XDMAC_CC25	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x06CC	XDMAC_CDS_MSP25	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x06D0	XDMAC_CSUS25	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x06D4	XDMAC_CDUS25	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x06D8	XDMAC_CTCS25	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x06DC ... 0x06DF	Reserved									
0x06E0	XDMAC_CIE26	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x06E4	XDMAC_CID26	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x06E8	XDMAC_CIM26	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x06EC	XDMAC_CIS26	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x06F0	XDMAC_CSA26	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x06F4	XDMAC_CDA26	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x06F8	XDMAC_CNDA26	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x06FC	XDMAC_CNDC26	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0700	XDMAC_CUBC26	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
...........continued										
0x0704	XDMAC_CBC26	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0708	XDMAC_CC26	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x070C	XDMAC_CDS_MSP26	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0710	XDMAC_CSUS26	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0714	XDMAC_CDUS26	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0718	XDMAC_CTCS26	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x071C ... 0x071F	Reserved									
0x0720	XDMAC_CIE27	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0724	XDMAC_CID27	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0728	XDMAC_CIM27	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x072C	XDMAC_CIS27	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0730	XDMAC_CSA27	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0734	XDMAC_CDA27	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0738	XDMAC_CNDA27	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x073C	XDMAC_CNDC27	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0740	XDMAC_CUBC27	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0744	XDMAC_CBC27	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0748	XDMAC_CC27	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x074C	XDMAC_CDS_MSP27	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0750	XDMAC_CSUS27	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0754	XDMAC_CDUS27	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0758	XDMAC_CTCS27	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x075C ... 0x075F	Reserved									
0x0760	XDMAC_CIE28	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0764	XDMAC_CID28	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0768	XDMAC_CIM28	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x076C	XDMAC_CIS28	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0770	XDMAC_CSA28	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0774	XDMAC_CDA28	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0778	XDMAC_CNDA28	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x077C	XDMAC_CNDC28	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0780	XDMAC_CUBC28	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0784	XDMAC_CBC28	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0788	XDMAC_CC28	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x078C	XDMAC_CDS_MSP28	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0790	XDMAC_CSUS28	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0794	XDMAC_CDUS28	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0798	XDMAC_CTCS28	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x079C ... 0x079F	Reserved									
0x07A0	XDMAC_CIE29	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x07A4	XDMAC_CID29	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x07A8	XDMAC_CIM29	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x07AC	XDMAC_CIS29	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x07B0	XDMAC_CSA29	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x07B4	XDMAC_CDA29	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x07B8	XDMAC_CNDA29	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x07BC	XDMAC_CNDC29	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x07C0	XDMAC_CUBC29	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x07C4	XDMAC_CBC29	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x07C8	XDMAC_CC29	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x07CC	XDMAC_CDS_MSP29	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x07D0	XDMAC_CSUS29	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x07D4	XDMAC_CDUS29	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x07D8	XDMAC_CTCS29	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x07DC ... 0x07DF	Reserved									
0x07E0	XDMAC_CIE30	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x07E4	XDMAC_CID30	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x07E8	XDMAC_CIM30	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x07EC	XDMAC_CIS30	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x07F0	XDMAC_CSA30	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x07F4	XDMAC_CDA30	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x07F8	XDMAC_CNDA30	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x07FC	XDMAC_CNDC30	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0800	XDMAC_CUBC30	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0804	XDMAC_CBC30	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0808	XDMAC_CC30	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x080C	XDMAC_CDS_MSP30	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0810	XDMAC_CSUS30	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0814	XDMAC_CDUS30	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0818	XDMAC_CTCS30	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							
0x081C ... 0x081F	Reserved									
0x0820	XDMAC_CIE31	31:24								
		23:16								
		15:8								
		7:0	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0824	XDMAC_CID31	31:24								
		23:16								
		15:8								
		7:0	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
0x0828	XDMAC_CIM31	31:24								
		23:16								
		15:8								
		7:0	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x082C	XDMAC_CIS31	31:24								
		23:16								
		15:8								
		7:0	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0830	XDMAC_CSA31	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0834	XDMAC_CDA31	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0838	XDMAC_CNDA31	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x083C	XDMAC_CNDC31	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0840	XDMAC_CUBC31	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0844	XDMAC_CBC31	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0848	XDMAC_CC31	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8				DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
0x084C	XDMAC_CDS_MSP31	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0850	XDMAC_CSUS31	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0854	XDMAC_CDUS31	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0858	XDMAC_CTCS31	31:24								
		23:16	TC[23:16]							
		15:8	TC[15:8]							
		7:0	TC[7:0]							

14.9.1 XDMAC Global Type Register

Name: XDMAC_GTYPE
Offset: 0x00
Property: Read-only

Reset: The register reset values depend on the instance of the XDMAC:

Instance	Reset Value
XDMAC0, XDMAC1	0x004A201F
XDMAC2	0x00020007

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		NB_REQ[6:0]						
Access		R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	FIFO_SZ[10:3]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	FIFO_SZ[2:0]			NB_CH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset								

Bits 22:16 – NB_REQ[6:0] Number of Peripheral Requests Minus One

Bits 15:5 – FIFO_SZ[10:0] Number of Bytes
The FIFO size is given by the formula: Number of bytes + 8.

Bits 4:0 – NB_CH[4:0] Number of Channels Minus One

14.9.2 XDMAC Global Configuration Register

Name: XDMAC_GCFG
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Notes:
1. WRxP fields: to maximize system performance and avoid system head of line blocking, verify that (WRHP+1)+(WRMP+1)+(WRLP+1) is less than or equal to the system buffering capability (i.e., the size of addresses and write data buffer in the system bus interconnect).
2. RDSG/RDxP fields: to maximize system performance and avoid system head of line blocking, verify that (RDSG+1)(RDHP+1)(RDMP+1)+(RDLP+1) is less than or equal to the system read buffering capability (i.e., the size of addresses buffer in the system bus interconnect).
3. WRxP, RDSG, RDxP fields are related to the RAQ and WAQ DDR queues, defined respectively in “Read Address Channel” and “Write Address Channel” in the section “Universal DDR Memory Controller (UDDRC)”.

Bit	31	30	29	28	27	26	25	24
	RDSG[3:0]				RDLP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RDMP[3:0]				RDHP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WRLP[3:0]				WRMP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WRHP[3:0]				CGDISIF	CGDISFIFO	CGDISPIPE	CGDISREG
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – RDSG[3:0] Read Queue Scatter Gather Outstanding Limit
RDSG+1 is the maximum number of active transactions for descriptor read access.

Bits 27:24 – RDLP[3:0] Read Queue Low Priority Outstanding Limit
RDLP+1 is the maximum number of active transactions for read access (memory) of a memory-to-memory transfer.

Bits 23:20 – RDMP[3:0] Read Queue Medium Priority Outstanding Limit
RDMP+1 is the maximum number of active transactions for read access (memory) of a memory-to-peripheral transfer.

Bits 19:16 – RDHP[3:0] Read Queue High Priority Outstanding Limit
RDHP+1 is the maximum number of active transactions for read access (peripheral) of a peripheral-to-memory transfer.

Bits 15:12 – WRLP[3:0] Write Queue Low Priority Outstanding Limit
WRLP+1 is the maximum number of active transactions for write access (memory) of a memory-to-memory transfer.

Bits 11:8 – WRMP[3:0] Write Queue Medium Priority Outstanding Limit
WRMP+1 is the maximum number of active transactions for write access (memory) of peripheral-to-memory transfer.

Bits 7:4 – WRHP[3:0] Write Queue High Priority Outstanding Limit
WRHP+1 is the maximum number of active transactions for write access (peripheral) of a memory-to-peripheral transfer.

Bit 3 – CGDISIF Bus Interface Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the system bus interface.
1	The automatic clock gating is disabled for the system bus interface.

Bit 2 – CGDISFIFO FIFO Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the main FIFO.
1	The automatic clock gating is disabled for the main FIFO.

Bit 1 – CGDISPIPE Pipeline Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the main pipeline.
1	The automatic clock gating is disabled for the main pipeline.

Bit 0 – CGDISREG Configuration Registers Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the configuration registers.
1	The automatic clock gating is disabled for the configuration registers.

14.9.3 XDMAC Global Weighted Arbiter Configuration Register

Name: XDMAC_GWAC
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Notes:

1. PW0/1: when a memory-to-peripheral transaction occurs, the peripheral write transaction may be asserted at the same time as a memory write transaction. The write port arbiter grants bus access to the highest transaction weight first, then the internal weight counter is decremented. When the internal weight counter reaches 0, it is reloaded with PWx. If PW0=0 and PW1 = 0, the arbitration method is round-robin.
2. PW2/3: When a peripheral-to-memory transaction occurs, the peripheral read transaction may be asserted at the same time as a memory read transaction. The Read Port Arbiter grants bus access to the highest transaction weight first, then the internal weight counter is decremented. When the internal weight counter reaches 0, it is reloaded with PWx. If PW2=0 and PW3 = 0, the arbitration method is round-robin.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PW3[3:0]				PW2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PW1[3:0]				PW0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – PW3[3:0] Pool Weight 3
Memory transaction weight for read port arbiter. This is exactly the peripheral read transaction weight of a memory-tomemory transfer.

Bits 11:8 – PW2[3:0] Pool Weight 2
Peripheral transaction weight for read port arbiter. This is exactly the peripheral read transaction weight of a peripheral-tomemory transfer.

Bits 7:4 – PW1[3:0] Pool Weight 1
Memory transaction weight for write port arbiter. This is exactly the memory write transaction weight of a memory-to-memory transfer.

Bits 3:0 – PW0[3:0] Pool Weight 0
Peripheral transaction weight for write port arbiter. This is exactly the peripheral write transaction weight of a memory-to-peripheral transfer.

14.9.4 XDMAC Global Interrupt Enable Register

Name: XDMAC_GIE
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IE31	IE30	IE29	IE28	IE27	IE26	IE25	IE24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IE23	IE22	IE21	IE20	IE19	IE18	IE17	IE16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – IEx XDMAC Channel x Interrupt Enable

Value	Description
0	This bit has no effect. The Channel x Interrupt Mask bit (XDMAC_GIM.IMx) is not modified.
1	The corresponding mask bit is set. The XDMAC Channel x Interrupt Status register (XDMAC_GIS) can generate an interrupt.

14.9.5 XDMAC Global Interrupt Disable Register

Name: XDMAC_GID
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – IDx XDMAC Channel x Interrupt Disable

Value	Description
0	This bit has no effect. The Channel x Interrupt Mask bit (XDMAC_GIM.IMx) is not modified.
1	The corresponding mask bit is reset. The Channel x Interrupt Status register interrupt (XDMAC_GIS) is masked.

14.9.6 XDMAC Global Interrupt Mask Register

Name: XDMAC_GIM
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – IMx XDMAC Channel x Interrupt Mask

Value	Description
0	This bit indicates that the channel x interrupt source is masked. The interrupt line is not raised.
1	This bit indicates that the channel x interrupt source is unmasked.

14.9.7 XDMAC Global Interrupt Status Register

Name: XDMAC_GIS
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ISx XDMAC Channel x Interrupt Status

Value	Description
0	This bit indicates that either the interrupt source is masked at the channel level or no interrupt is pending for channel x.
1	This bit indicates that an interrupt is pending for the channel x.

14.9.8 XDMAC Global Channel Enable Register

Name: XDMAC_GE
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ENx XDMAC Channel x Enable

Value	Description
0	This bit has no effect.
1	Enables channel n. This operation is permitted if the Channel x Status bit (XDMAC_GS.STx) was read as '0'.

14.9.9 XDMAC Global Channel Disable Register

Name: XDMAC_GD
Offset: 0x20
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	DI31	DI30	DI29	DI28	DI27	DI26	DI25	DI24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	DI23	DI22	DI21	DI20	DI19	DI18	DI17	DI16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DIx XDMAC Channel x Disable

Value	Description
0	This bit has no effect.
1	Disables channel x.

14.9.10 XDMAC Global Channel Status Register

Name: XDMAC_GS
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ST31	ST30	ST29	ST28	ST27	ST26	ST25	ST24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ST23	ST22	ST21	ST20	ST19	ST18	ST17	ST16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – STx XDMAC Channel x Status

Value	Description
0	This bit indicates that the channel x is disabled.
1	This bit indicates that the channel x is enabled. If a channel disable request is issued, this bit remains asserted until pending transaction is completed.

14.9.11 XDMAC Global Channel Read Suspend Register

Name: XDMAC_GRS
Offset: 0x30
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	RS31	RS30	RS29	RS28	RS27	RS26	RS25	RS24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – RSx XDMAC Channel x Read Suspend

Value	Description
0	No effect.
1	The source requests for channel n are no longer serviced by the system scheduler.

14.9.12 XDMAC Global Channel Write Suspend Register

Name: XDMAC_GWS
Offset: 0x38
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	WS31	WS30	WS29	WS28	WS27	WS26	WS25	WS24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WS23	WS22	WS21	WS20	WS19	WS18	WS17	WS16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – WSx XDMAC Channel x Write Suspend

Value	Description
0	No effect.
1	Destination requests are no longer routed to the scheduler.

14.9.13 XDMAC Global Channel Read Write Suspend Register

Name: XDMAC_GRWS
Offset: 0x40
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	RWS31	RWS30	RWS29	RWS28	RWS27	RWS26	RWS25	RWS24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	RWS23	RWS22	RWS21	RWS20	RWS19	RWS18	RWS17	RWS16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – RWSx XDMAC Channel x Read Write Suspend

Value	Description
0	No effect.
1	Read and write requests are suspended.

14.9.14 XDMAC Global Channel Read Write Resume Register

Name: XDMAC_GRWR
Offset: 0x44
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	RWR31	RWR30	RWR29	RWR28	RWR27	RWR26	RWR25	RWR24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RWR23	RWR22	RWR21	RWR20	RWR19	RWR18	RWR17	RWR16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RWR15	RWR14	RWR13	RWR12	RWR11	RWR10	RWR9	RWR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RWR7	RWR6	RWR5	RWR4	RWR3	RWR2	RWR1	RWR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – RWRx XDMAC Channel x Read Write Resume

Value	Description
0	No effect.
1	Read and write requests are serviced.

14.9.15 XDMAC Global Channel Read Suspend Status Register

Name: XDMAC_GRSS
Offset: 0x28
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RSS31	RSS30	RSS29	RSS28	RSS27	RSS26	RSS25	RSS24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RSS23	RSS22	RSS21	RSS20	RSS19	RSS18	RSS17	RSS16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RSS15	RSS14	RSS13	RSS12	RSS11	RSS10	RSS9	RSS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RSS7	RSS6	RSS5	RSS4	RSS3	RSS2	RSS1	RSS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – RSS XDMAC Channel x Read Suspend Status

Value	Description
0	The read channel is not suspended.
1	The source requests for channel x are no longer serviced by the system scheduler.

14.9.16 XDMAC Global Channel Write Suspend Status Register

Name: XDMAC_GWSS
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WSS31	WSS30	WSS29	WSS28	WSS27	WSS26	WSS25	WSS24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WSS23	WSS22	WSS21	WSS20	WSS19	WSS18	WSS17	WSS16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WSS15	WSS14	WSS13	WSS12	WSS11	WSS10	WSS9	WSS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WSS7	WSS6	WSS5	WSS4	WSS3	WSS2	WSS1	WSS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – WSS XDMAC Channel x Write Suspend Status

Value	Description
0	The write channel is not suspended.
1	The source requests for channel x are no longer serviced by the system scheduler.

14.9.17 XDMAC Global Channel Read Resume Register

Name: XDMAC_GRR
Offset: 0x34
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	RR31	RR30	RR29	RR28	RR27	RR26	RR25	RR24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RR23	RR22	RR21	RR20	RR19	RR18	RR17	RR16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RR15	RR14	RR13	RR12	RR11	RR10	RR9	RR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RR7	RR6	RR5	RR4	RR3	RR2	RR1	RR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – RR
XDMAC Channel x Read Resume

Value	Description
0	No effect
1	The source requests for channel x are serviced by the system scheduler.

14.9.18 XDMAC Global Channel Write Resume Register

Name: XDMAC_GWR
Offset: 0x3C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	WR31	WR30	WR29	WR28	WR27	WR26	WR25	WR24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WR23	WR22	WR21	WR20	WR19	WR18	WR17	WR16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WR15	WR14	WR13	WR12	WR11	WR10	WR9	WR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WR7	WR6	WR5	WR4	WR3	WR2	WR1	WR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – WR XDMAC Channel x Write Resume

Value	Description
0	No effect.
1	Destination requests are serviced and routed to the scheduler.

14.9.19 XDMAC Global Channel Software Request Register

Name: XDMAC_GSWR
Offset: 0x48
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	SWREQ31	SWREQ30	SWREQ29	SWREQ28	SWREQ27	SWREQ26	SWREQ25	SWREQ24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	SWREQ23	SWREQ22	SWREQ21	SWREQ20	SWREQ19	SWREQ18	SWREQ17	SWREQ16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	SWREQ15	SWREQ14	SWREQ13	SWREQ12	SWREQ11	SWREQ10	SWREQ9	SWREQ8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SWREQx XDMAC Channel x Software Request

Value	Description
0	No effect.
1	Requests a DMA transfer for channel x.

14.9.20 XDMAC Global Channel Software Request Status Register

Name: XDMAC_GSWS
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SWRS31	SWRS30	SWRS29	SWRS28	SWRS27	SWRS26	SWRS25	SWRS24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SWRS23	SWRS22	SWRS21	SWRS20	SWRS19	SWRS18	SWRS17	SWRS16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SWRS15	SWRS14	SWRS13	SWRS12	SWRS11	SWRS10	SWRS9	SWRS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWRS7	SWRS6	SWRS5	SWRS4	SWRS3	SWRS2	SWRS1	SWRS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SWRSx XDMAC Channel x Software Request Status

Value	Description
0	Channel x source request is serviced.
1	Channel x source request is pending.

14.9.21 XDMAC Global Channel Software Flush Request Register

Name: XDMAC_GSWF
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SWF31	SWF30	SWF29	SWF28	SWF27	SWF26	SWF25	SWF24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SWF23	SWF22	SWF21	SWF20	SWF19	SWF18	SWF17	SWF16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SWFx XDMAC Channel x Software Flush Request

Value	Description
0	No effect.
1	Requests a DMA transfer flush for channel x. This bit is only relevant when the transfer is source peripheral synchronized.

14.9.22 XDMAC Channel x Interrupt Enable Register [x=0..31]

Name: XDMAC_CIE
Offset: 0x60 + n*0x40 [n=0..31]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TCIE	ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – TCIE Transfer Count Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enables transfer count overflow error interrupt.

Bit 6 – ROIE Request Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enables request overflow error interrupt.

Bit 5 – WBIE Write Bus Error Interrupt Enable

Value	Description
0	No effect.
1	Enables write bus error interrupt.

Bit 4 – RBIE Read Bus Error Interrupt Enable

Value	Description
0	No effect.
1	Enables read bus error interrupt.

Bit 3 – FIE End of Flush Interrupt Enable

Value	Description
0	No effect.
1	Enables end of flush interrupt.

Bit 2 – DIE End of Disable Interrupt Enable

Value	Description
0	No effect.
1	Enables end of disable interrupt.

Bit 1 – LIE End of Linked List Interrupt Enable

Value	Description
0	No effect.
1	Enables end of linked list interrupt.

Bit 0 – BIE End of Block Interrupt Enable

Value	Description
0	No effect.
1	Enables end of block interrupt.

14.9.23 XDMAC Channel x Interrupt Disable Register [x = 0..31]

Name: XDMAC_CID
Offset: 0x64 + n*0x40 [n=0..31]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TCID	ROID	WBEID	RBEID	FID	DID	LID	BID
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – TCID Transfer Count Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disables transfer count overflow error interrupt.

Bit 6 – ROID Request Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disables request overflow error interrupt.

Bit 5 – WBEID Write Bus Error Interrupt Disable

Value	Description
0	No effect.
1	Disables bus error interrupt.

Bit 4 – RBEID Read Bus Error Interrupt Disable

Value	Description
0	No effect.
1	Disables bus error interrupt.

Bit 3 – FID End of Flush Interrupt Disable

Value	Description
0	No effect.
1	Disables end of flush interrupt.

Bit 2 – DID End of Disable Interrupt Disable

Value	Description
0	No effect.
1	Disables end of disable interrupt.

Bit 1 – LID End of Linked List Interrupt Disable

Value	Description
0	No effect.
1	Disables end of linked list interrupt.

Bit 0 – BID End of Block Interrupt Disable

Value	Description
0	No effect.
1	Disables end of block interrupt.

14.9.24 XDMAC Channel x Interrupt Mask Register [x = 0..31]

Name: XDMAC_CIM
Offset: 0x68 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TCIM	ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – TCIM Transfer Count Overflow Error Interrupt Mask

Value	Description
0	Transfer count overflow interrupt is masked.
1	Transfer count overflow interrupt is activated.

Bit 6 – ROIM Request Overflow Error Interrupt Mask

Value	Description
0	Request overflow interrupt is masked.
1	Request overflow interrupt is activated.

Bit 5 – WBEIM Write Bus Error Interrupt Mask

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 4 – RBEIM Read Bus Error Interrupt Mask

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 3 – FIM End of Flush Interrupt Mask

Value	Description
0	End of flush interrupt is masked.
1	End of flush interrupt is activated.

Bit 2 – DIM End of Disable Interrupt Mask

Value	Description
0	End of disable interrupt is masked.
1	End of disable interrupt is activated.

Bit 1 – LIM End of Linked List Interrupt Mask

Value	Description
0	End of linked list interrupt is masked.
1	End of linked list interrupt is activated.

Bit 0 – BIM End of Block Interrupt Mask

Value	Description
0	Block interrupt is masked.
1	Block interrupt is activated.

14.9.25 XDMAC Channel x Interrupt Status Register [x = 0..31]

Name: XDMAC_CIS
Offset: 0x6C + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TCIS	ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – TCIS Transfer Count Overflow Interrupt Status

Value	Description
0	Transfer count overflow has not occurred.
1	At least one transfer count overflow has been detected since the last read of the Status register.

Bit 6 – ROIS Request Overflow Error Interrupt Status

Value	Description
0	Overflow condition has not occurred.
1	Overflow condition has occurred at least once. (This information is only relevant for peripheral synchronized transfers.)

Bit 5 – WBEIS Write Bus Error Interrupt Status

Value	Description
0	Write bus error condition has not occurred.
1	At least one bus error has been detected in a write access since the last read of the Status register.

Bit 4 – RBEIS Read Bus Error Interrupt Status

Value	Description
0	Read bus error condition has not occurred.
1	At least one bus error has been detected in a read access since the last read of the Status register.

Bit 3 – FIS End of Flush Interrupt Status

Value	Description
0	End of flush condition has not occurred.
1	End of flush condition has occurred since the last read of the Status register.

Bit 2 – DIS End of Disable Interrupt Status

Value	Description
0	End of disable condition has not occurred.

Value	Description
1	End of disable condition has occurred since the last read of the Status register.

Bit 1 – LIS End of Linked List Interrupt Status

Value	Description
0	End of linked list condition has not occurred.
1	End of linked list condition has occurred since the last read of the Status register.

Bit 0 – BIS End of Block Interrupt Status

Value	Description
0	End of block interrupt has not occurred.
1	End of block interrupt has occurred since the last read of the Status register.

14.9.26 XDMAC Channel x Source Address Register [x = 0..31]

Name: XDMAC_CSA
Offset: 0x70 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SA[31:0] Channel x Source Address
Program this register with the source address of the DMA transfer.

14.9.27 XDMAC Channel x Destination Address Register [x = 0..31]

Name: XDMAC_CDA
Offset: 0x74 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DA[31:0] Channel x Destination Address
Program this register with the destination address of the DMA transfer.

14.9.28 XDMAC Channel x Next Descriptor Address Register [x = 0..31]

Name: XDMAC_CNDA
Offset: 0x78 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NDA[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NDA[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NDA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NDA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – NDA[29:0] Channel x Next Descriptor Address
The 30-bit width of the NDA field represents the next descriptor address range 31:2. The descriptor is word-aligned and the two least significant register bits 1:0 are ignored.

14.9.29 XDMAC Channel x Next Descriptor Control Register [x = 0..31]

Name: XDMAC_CNDC
Offset: 0x7C + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:5 – QOS[1:0] Channel Quality Of Service level
This field indicates the current quality of service level for the channel. Refer to the section "MATRIX".

Bits 4:3 – NDVIEW[1:0] Channel x Next Descriptor View

Value	Name	Description
0	NDV0	Next Descriptor View 0
1	NDV1	Next Descriptor View 1
2	NDV2	Next Descriptor View 2
3	NDV3	Next Descriptor View 3

Bit 2 – NDDUP Channel x Next Descriptor Destination Update
0 (): .
1 ():

Value	Name	Description
0	DST_PARAMS_UNCHANGED	Destination parameters remain unchanged.
1	DST_PARAMS_UPDATED	Destination parameters are updated when the descriptor is retrieved.

Bit 1 – NDSUP Channel x Next Descriptor Source Update

Value	Name	Description
0	SRC_PARAMS_UNCHANGED	Source parameters remain unchanged.
1	SRC_PARAMS_UPDATED	Source parameters are updated when the descriptor is retrieved.

Bit 0 – NDE Channel x Next Descriptor Enable

Value	Name	Description
0	DSCR_FETCH_DIS	Descriptor fetch is disabled.
1	DSCR_FETCH_EN	Descriptor fetch is enabled.

14.9.30 XDMAC Channel x Microblock Control Register [x = 0..31]

Name: XDMAC_CUBC
Offset: 0x80 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	UBLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UBLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UBLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – UBLEN[23:0] Channel x Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

14.9.31 XDMAC Channel x Block Control Register [x = 0..31]

Name: XDMAC_CBC
Offset: 0x84 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					BLEN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – BLEN[11:0] Channel x Block Length
The length of the block is (BLEN+1) microblocks.

14.9.32 XDMAC Channel x Configuration Register [x = 0..31]

Name: XDMAC_CC
Offset: 0x88 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		PERID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
Access	R	R	R		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0
Bit	15	14	13	12	11	10	9	8
				DWIDTH[1:0]		CSIZE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MEMSET	SWREQ	PROT	DSYNC		MBSIZE[1:0]		TYPE
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 30:24 – PERID[6:0] Channel x Peripheral Hardware Request Line Identifier
This field contains the peripheral hardware request line identifier. PERID refers to identifiers defined in "DMA Controller Peripheral Connections".
Note: When a memory-to-memory transfer is performed, configure PERID to 0x7F.

Bit 23 – WRIP Write in Progress

Value	Name	Description
0	DONE	No active write transaction on the bus.
1	IN_PROGRESS	A write transaction is in progress.

Bit 22 – RDIP Read in Progress

Value	Name	Description
0	DONE	No active read transaction on the bus.
1	IN_PROGRESS	A read transaction is in progress.

Bit 21 – INITD Channel Initialization Done
When set to 0, XDMAC_CUBC.UBLEN and XDMAC_CNDA.NDA field values are unreliable each time a descriptor is being updated. See 14.8. XDMAC Software Requirements.

Value	Name	Description
0	IN_PROGRESS	Channel initialization is in progress.
1	TERMINATED	Channel initialization is completed.

Bits 19:18 – DAM[1:0] Channel x Destination Addressing Mode

Value	Name	Description
0	FIXED_AM	The address remains unchanged.
1	INCREMENTED_AM	The addressing mode is incremented (the increment size is set to the data size).
2	UBS_AM	The microblock stride is added at the microblock boundary.
3	UBS_DS_AM	The microblock stride is added at the microblock boundary; the data stride is added at the data boundary.

Bits 17:16 – SAM[1:0] Channel x Source Addressing Mode

Value	Name	Description
0	FIXED_AM	The address remains unchanged.
1	INCREMENTED_AM	The addressing mode is incremented (the increment size is set to the data size).
2	UBS_AM	The microblock stride is added at the microblock boundary.
3	UBS_DS_AM	The microblock stride is added at the microblock boundary, the data stride is added at the data boundary.

Bits 12:11 – DWIDTH[1:0] Channel x Data Width

Value	Name	Description
0	BYTE	The data size is set to 8 bits
1	HALFWORD	The data size is set to 16 bits
2	WORD	The data size is set to 32 bits
3	DWORD	The data size is set to 64 bits

Bits 10:8 – CSIZE[2:0] Channel x Chunk Size

Value	Name	Description
0	CHK_1	1 data transferred
1	CHK_2	2 data transferred
2	CHK_4	4 data transferred
3	CHK_8	8 data transferred
4	CHK_16	16 data transferred

Bit 7 – MEMSET Channel x Fill Block of Memory

Value	Name	Description
0	NORMAL_MODE	Memset is not activated.
1	HW_MODE	Sets the block of memory pointed by DA field to the specified value. This operation is performed on 8-, 16- or 32-bit basis.

Bit 6 – SWREQ Channel x Software Request Trigger

Value	Name	Description
0	HWR_CONNECTED	Hardware request line is connected to the peripheral request line.
1	SWR_CONNECTED	Software request is connected to the peripheral request line.

Bit 5 – PROT Channel x Protection

When a descriptor is loaded, the PROT bit cannot be modified. If PROT=0 for a channel, the configuration and status registers of this channel cannot be modified by unsecure software.

Value	Name	Description
0	SEC	Channel is secured.
1	UNSEC	Channel is unsecured.

Bit 4 – DSYNC Channel x Synchronization

Value	Name	Description
0	PER2MEM	Peripheral-to-memory transfer
1	MEM2PER	Memory-to-peripheral transfer

Bits 2:1 – MBSIZE[1:0] Channel x Memory Burst Size

Value	Name	Description
0	SINGLE	The memory burst size is set to one.
1	FOUR	The memory burst size is set to four.
2	EIGHT	The memory burst size is set to eight.
3	SIXTEEN	The memory burst size is set to sixteen.

Bit 0 – TYPE Channel x Transfer Type

Value	Name	Description
0	MEM_TRAN	Self-triggered mode (memory-to-memory transfer)
1	PER_TRAN	Synchronized mode (peripheral-to-memory or memory-to-peripheral transfer)

14.9.33 XDMAC Channel x Data Stride Memory Set Pattern Register [x = 0..31]

Name: XDMAC_CDS_MSP
Offset: 0x8C + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DDS_MSP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DDS_MSP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDS_MSP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDS_MSP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DDS_MSP[15:0] Channel x Destination Data Stride or Memory Set Pattern
When XDMAC_CCx.MEMSET = 0, this field indicates the destination data stride.
Number of bytes for the data stride of channel x (two's complement). If the field is set to zero the data is contiguous (see Data Striding Diagram).
The DDS_MSP field is only relevant when XDMAC_CCx.SAM=UBS_DS_AM.
When XDMAC_CCx.MEMSET = 1, this field indicates the memory set pattern.

Bits 15:0 – SDS_MSP[15:0] Channel x Source Data Stride or Memory Set Pattern
When XDMAC_CCx.MEMSET = 0, this field indicates the source data stride.
Number of bytes for the data stride of channel x (two's complement). If the field is set to zero the data is contiguous (see Data Striding Diagram).
The SDS_MSP field is only relevant when XDMAC_CCx.SAM=UBS_DS_AM.
When XDMAC_CCx.MEMSET = 1, this field indicates the memory set pattern.

14.9.34 XDMAC Channel x Source Microblock Stride Register [x = 0..31]

Name: XDMAC_CSUS
Offset: 0x90 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	SUBS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	SUBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SUBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – SUBS[23:0] Channel x Source Microblock Stride
Number of bytes for the microblock stride for channel x (two's complement). If the field is set to zero the data is contiguous (see Figure 14-2).
The SUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

14.9.35 XDMAC Channel x Destination Microblock Stride Register [x = 0..31]

Name: XDMAC_CDUS
Offset: 0x94 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DUBS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DUBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DUBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – DUBS[23:0] Channel x Destination Microblock Stride
Number of bytes for the microblock stride for channel x (two's complement). If the field is set to zero the data is contiguous (see Figure 14-2).
The DUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

14.9.36 XDMAC Channel x Transfer Count Status Register [x = 0..XDMAC_NB_CH-131]

Name: XDMAC_CTCS
Offset: 0x98 + n*0x40 [n=0..31]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

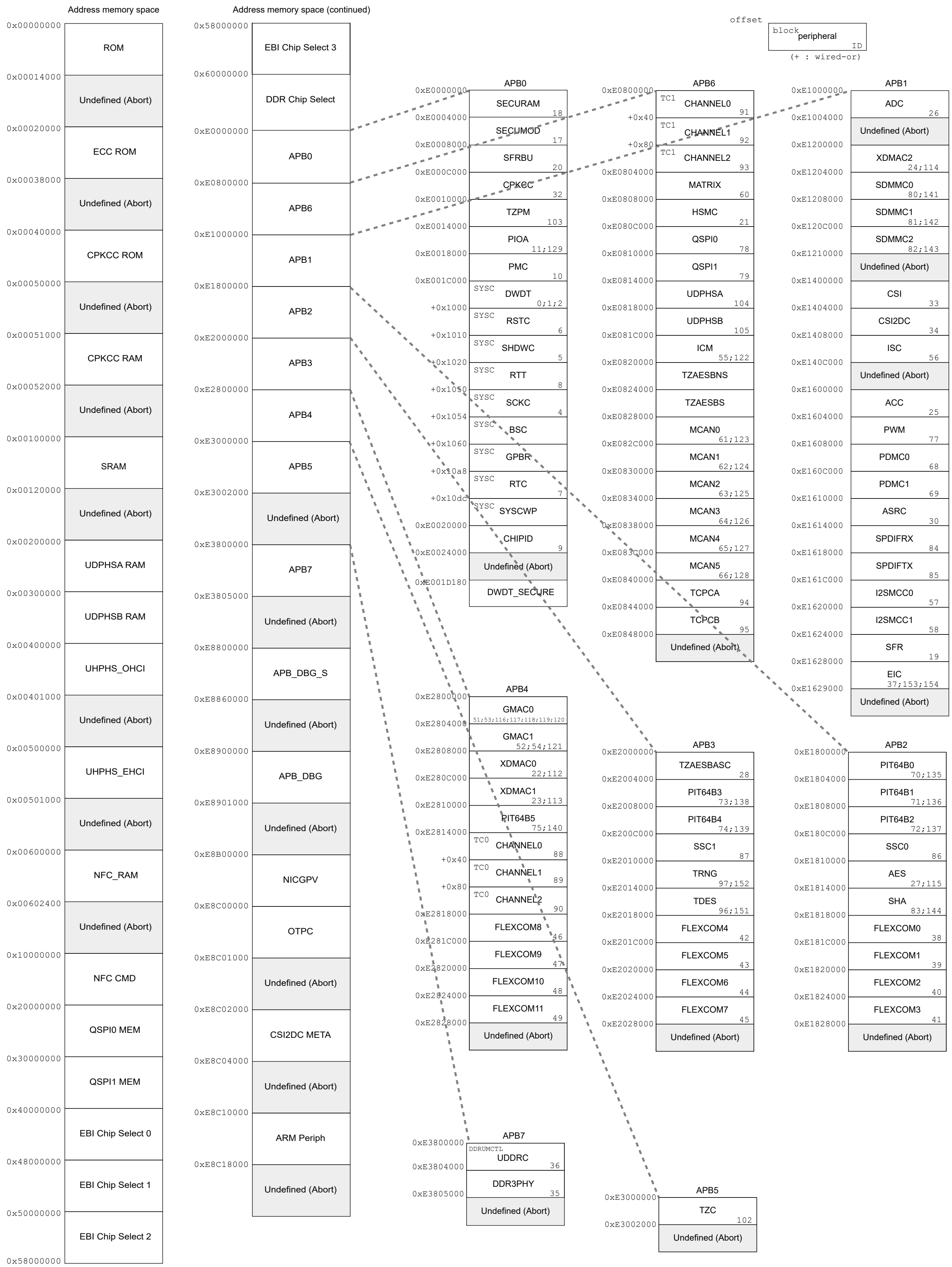
Bit	7	6	5	4	3	2	1	0
	TC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – TC[23:0] Channel x Transfer Count
Number of bytes written by the XDMA since the last read of the Transfer Count Status register. The field does not indicate the number of bytes written to memory.
To get the number of bytes already written in memory at any time during peripheral-to-memory transfer, use the following software procedure:

1. Read the XDMAC_CTCSx.TC field to sample the number of bytes received by the XDMA.
2. Perform a software flush of the channel by writing one to the XDMAC_GSWF register. This will push bytes from the internal DMA FIFO to the external memory and return an interrupt when the bytes can be read by the CPU.

15. MEMORIES

Figure 15-1. Memory Mapping



15.1 Embedded Memories

15.1.1 Internal SRAM

The device embeds a total of 128 Kbytes of high-speed SRAM. The SRAM is always accessible at address 0x00100000.

The device embeds 5 Kbytes of SRAM (SECURAM) always accessible at address 0xE0000000. The SECURAM is used by OTPC and SECUMOD.

15.1.2 Internal ROM

The ROM contents are:

- ROM0: Secure bootloader (80 Kbytes) at offset 0
- ROM1: BCH tables to compute NAND Flash ECC (96 Kbytes) at address 0X00002000
- ROM2: Crypto-libraries CPKCC (40 Kbytes) at address 0X00040000

By construction, ROMs are only accessible by the CPU with the restriction below:

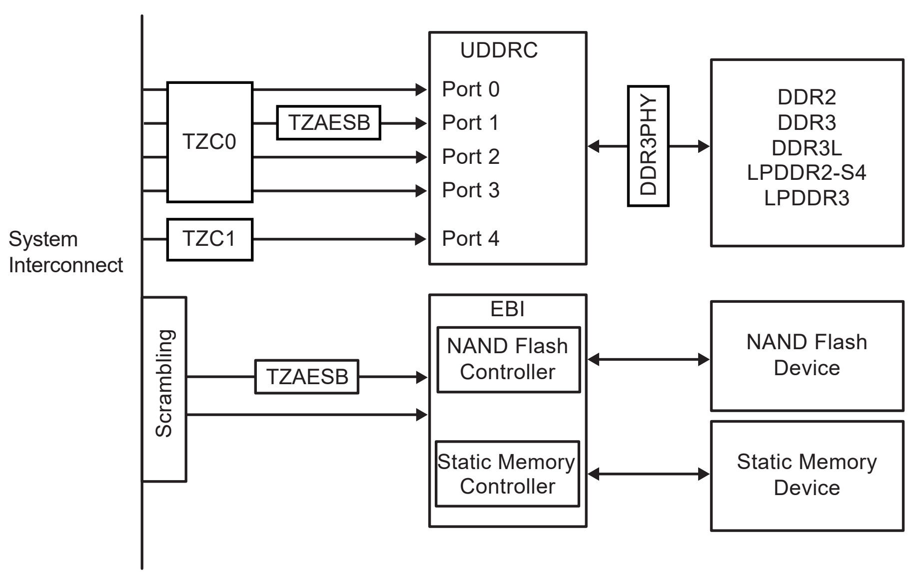
- ROM0: accessible only in Secure mode and when executed after reset. This memory is hidden before running code in internal SRAM.
- ROM1: always accessible in Secure and Non-secure modes
- ROM2: always accessible in Secure mode

15.2 External Memories

The device features:

- A Universal DDR-SDRAM Controller (UDDRC), supporting DDR2, DDR3, DDR3L, LPDDR2-S4 and LPDDR3 devices.
- An External Bus Interface (EBI) that embeds a NAND Flash controller and a Static Memory Controller (HSMC). The HSMC supports Static Memories and MLC/SLC NAND Flash. It embeds MultiBit ECC correction (PMECC). The HSMC buses can be scrambled and can be connected to an AES encryption/decryption engine.

Figure 15-2. External Memory Controllers



For more details, refer to I/O AC Characteristics and System Interconnect and Security (SIS).

15.2.1 Universal DDR-SDRAM Controller (UDDRC)

The UDDRC is a 5-port memory controller supporting DDR-SDRAM and low-power DDR devices. Data transfers are performed through a 16-bit data bus on one chip select. The memory interface operates with a 1.8V power supply for DDR2, 1.5V for DDR3, 1.35V for DDR3L, and 1.2V for LPDDR2 and LPDDR3.

For full details, refer to the sections Universal DDR-SDRAM Controller (UDDRC) and DDR/LPDDR Physical Interface (DDR3PHY).

15.2.2 External Bus Interface (EBI)

The External Bus Interface is designed to ensure the successful data transfer between several external devices and the processor. The EBI of the device consists of a Static Memory Controller (SMC) and a NAND Flash Controller (NFC).

For full details, refer to the section Static Memory Controller (SMC).

15.2.2.1 Description

The External Bus Interface is designed to ensure the successful data transfer between several external devices and the Arm processor-based device. The device EBI consists of a Static Memory Controller (SMC).

15.3 Product Dependencies

15.3.1 Clocks

Memory clocks are not controlled by the PMC.

- ROM and CPKCC are on the CPU System and Security (CSS) matrix and therefore clocked by MCK0.
- SECURAM is located on CSS and clocked by MCK0 divided by 32.
- SRAM, SMC and NFC_RAM are located on the AHB System (HSS) matrix, clocked by MCK1.
- UDDRC and DDR3PHY are clocked by MCK2 and hard-wired to the DDRPLLCK.

Note: The MCK0 frequency is directly related to the CPU clock, so any change on the CPU clock impacts MCK0.

⚠ CAUTION MCK3 must be started during UDDRC and DDR3PHY initialization.

15.3.2 Interrupts

Refer to the table Peripheral Identifiers.

15.3.3 Reset

SMC and NFC are connected to the processor and peripherals reset line.

UDDRC and DDR3PHY have separated reset lines, controlled in RSTC_ GRSTR.

15.3.4 I/Os

DDR-SDRAM I/Os are high-speed specific I/Os powered by VDDIODDR. These can be programmed in Retention state in order to maintain their state in Backup Self-refresh mode.

15.4 Special Functions in SFR/SFRBU

- SFR_UDDRC.DIS_DECERR, to be set during the debug phase to avoid UDDRC decode errors with the debugger
- SFRBU_DDRPWR.RETENTION to control DDR-SDRAM I/O Retention state

16. Static Memory Controller (SMC)

16.1 Description

This Static Memory Controller (SMC) is capable of handling several types of external memory and peripheral devices, such as SRAM, PSRAM, PROM, EPROM, EEPROM, LCD Module, NOR Flash and NAND Flash.

The SMC generates the signals that control the access to external memory devices or peripheral devices. It has 4 Chip Selects and a 26-bit address bus. The 16-bit data bus can be configured to interface with 8- or 16-bit external devices. Separate read and write control signals allow for direct memory and peripheral interfacing. Read and write signal waveforms are fully configurable.

The SMC can manage wait requests from external devices to extend the current access. The SMC is provided with an automatic Slow Clock mode. In Slow Clock mode, it switches from user-programmed waveforms to slow-rate specific waveforms on read and write signals.

The SMC embeds a NAND Flash Controller (NFC). The NFC can handle automatic transfers, sending the commands and address cycles to the NAND Flash and transferring the contents of the page (for read and write) to the NFC SRAM. It minimizes the CPU overhead.

The SMC embeds a programmable binary BCH encoder/decoder that generates redundancy information at encoding for both SLC and MLC NAND devices. It supports redundancy for correction of 2, 4, 8, 12, 24, or 32 errors per sector of 512 or 1024 bytes.

The External Data Bus can be scrambled/unscrambled by means of user keys.

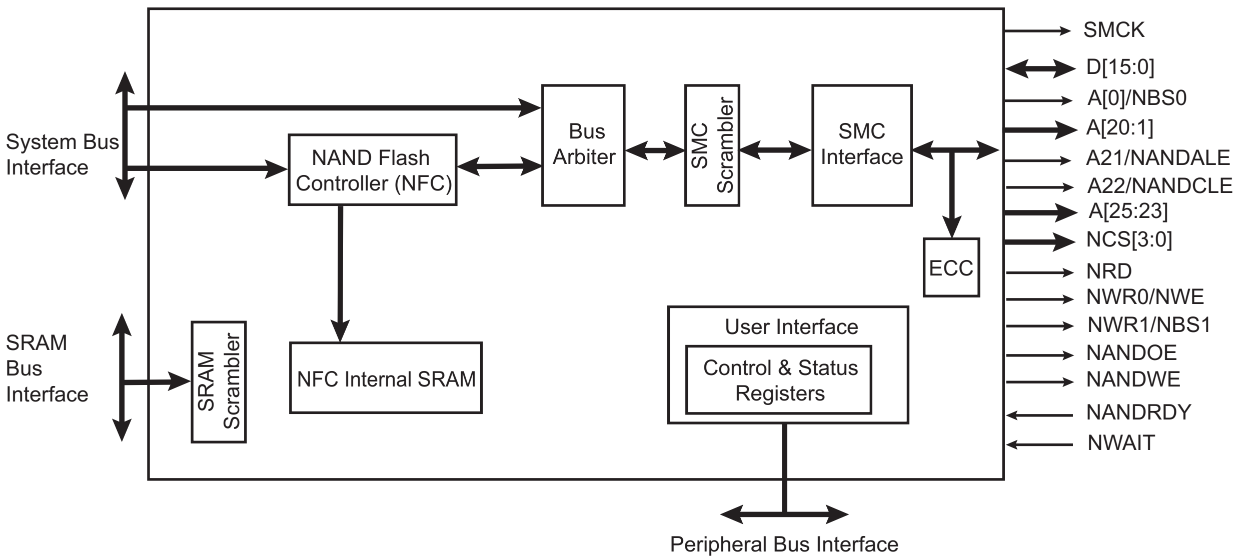
16.2 Embedded Characteristics

- 128-Mbyte Address Space per Chip Select
- 8- or 16-bit Data Bus
- Word, Halfword, Byte Transfers
- Byte Write or Byte Select Lines
- Programmable Setup, Pulse and Hold Time for Read Signals per Chip Select
- Programmable Setup, Pulse and Hold Time for Write Signals per Chip Select
- Programmable Data Float Time per Chip Select
- External Data Bus Scrambling/Unscrambling Function
- External Wait Request
- Automatic Switch to Slow Clock Mode
- Hardware Configurable Number of Chip Selects from 1 to 4
- Programmable Timing on a per Chip Select Basis
- NAND Flash Controller Supporting NAND Flash with Multiplexed Data/Address Buses
- Supports SLC and MLC NAND Flash Technology
- Supports NAND Flash Devices with 8 or 16-bit Data Paths
- Multibit Error Correcting Code (ECC) supporting NAND Flash devices with 8-bit only Data Path
- ECC Algorithm Based on Binary Shortened Bose, Chaudhuri and Hocquenghem (BCH) Codes
- Programmable Error Correcting Capability: 2, 4, 8, 12, 24 and 32 bits of Errors per Block
- 9 Kbytes NFC SRAM (NFC_RAM)
- Programmable Block Size: 512 bytes or 1024 bytes
- Programmable Number of Blocks per Page: 1, 2, 4 or 8 Blocks of Data per Page
- Programmable Spare Area Size up to 512 bytes
- Supports Spare Area ECC Protection
- Supports 8 Kbytes Page Size Using 1024 bytes/block and 4 Kbytes Page Size Using 512 bytes/block
- Multibit Error Detection Is Interrupt Driven

- Provides Hardware Acceleration for Determining Roots of Polynomials Defined over a Finite Field
- Programmable Finite Field GF(2^13) or GF(2^14)
- Finds Roots of Error-locator Polynomial
- Programmable Number of Roots
- Register Write Protection

16.3 Block Diagram

Figure 16-1. SMC Block Diagram



16.4 I/O Lines Description

Table 16-1. I/O Lines Description

Name	Description	Type	Active Level
SMCK	Static Memory Controller Clock	Output	–
NCS[3:0]	Static Memory Controller Chip Select Lines	Output	Low
NRD	Read Signal	Output	Low
NWR0/NWE	Write 0/Write Enable Signal	Output	Low
A0/NBS0	Address Bit 0/Byte 0 Select Signal	Output	Low
NWR1/NBS1	Write 1/Byte 1 Select Signal	Output	Low
A[25:1]	Address Bus	Output	–
D[15:0]	Data Bus	I/O	–
NWAIT	External Wait Signal	Input	Low
NANDRDY	NAND Flash Ready/Busy	Input	–
NANDWE	NAND Flash Write Enable	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDALE	NAND Flash Address Latch Enable	Output	–

...........continued

Name	Description	Type	Active Level
NANDCLE	NAND Flash Command Latch Enable	Output	–

16.5 Multiplexed Signals

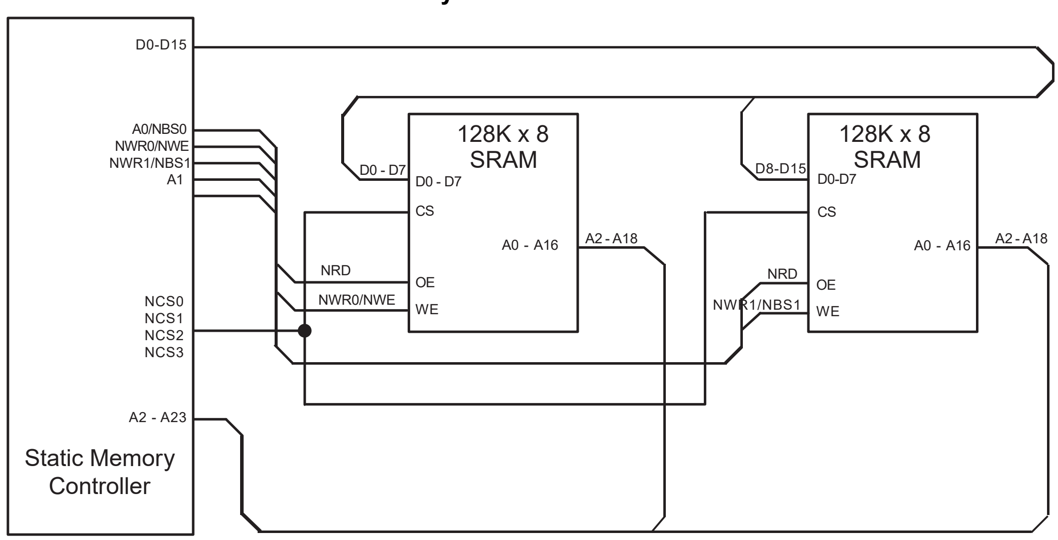
Table 16-2. SMC Multiplexed Signals

Multiplexed Signals		Related Function
NWR0	NWE	Byte-write or Byte-select access, see Byte Write Access and Byte Select Access
A0	NBS0	8-bit or 16-bit data bus, see Data Bus Width
A22	NANDCLE	NAND Flash Command Latch Enable
A21	NANDALE	NAND Flash Address Latch Enable
NWR1	NBS1	Byte-write or Byte-select access, see Byte Write Access and Byte Select Access
A1	–	8-/16-bit data bus, see Data Bus Width Byte-write or Byte-select access, see Byte Write Access and Byte Select Access

16.6 Application Example

16.6.1 Hardware Interface

Figure 16-2. SMC Connections to Static Memory Devices



16.7 Product Dependencies

16.7.1 I/O Lines

The pins used for interfacing the Static Memory Controller are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the Static Memory Controller pins to their peripheral function. If I/O lines of the SMC are not used by the application, they can be used for other purposes by the PIO controller.

16.7.2 Power Management

The SMC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the SMC clock.

16.7.3 Interrupt Sources

The SMC has an interrupt line connected to the interrupt controller. Handling the SMC interrupt requires programming the interrupt controller before configuring the SMC.

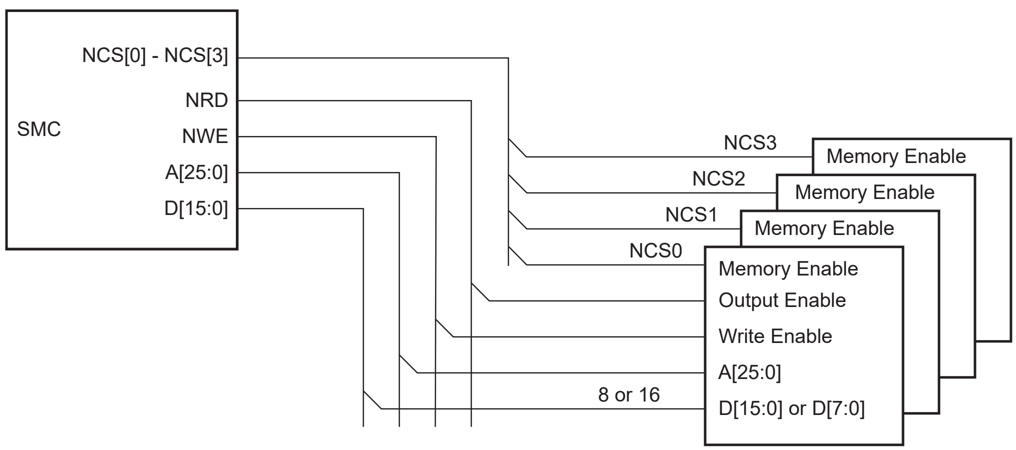
16.8 External Memory Mapping

The SMC provides up to 26 address lines, A[25:0]. This allows each chip select line to address up to 128 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 128 Mbytes, it wraps around and appears to be repeated within this space. The SMC correctly handles any valid access to the memory device within the page. See the figure below.

A[25:0] is only significant for 8-bit memory; A[25:1] is used for 16-bit memory.

Figure 16-3. Memory Connections for External Devices



16.9 Connection to External Devices

16.9.1 Data Bus Width

A data bus width of 8 or 16 bits can be selected for each chip select. This option is controlled by the bit DBW in the Mode register (HSMC_MODE) for the corresponding chip select.

The figure Memory Connection for an 8-bit Data Bus shows how to connect a 512 KB x 8-bit memory on NCS2. The figure Memory Connection for a 16-bit Data Bus shows how to connect a 512 KB x 16-bit memory on NCS2.

Figure 16-4. Memory Connection for an 8-bit Data Bus

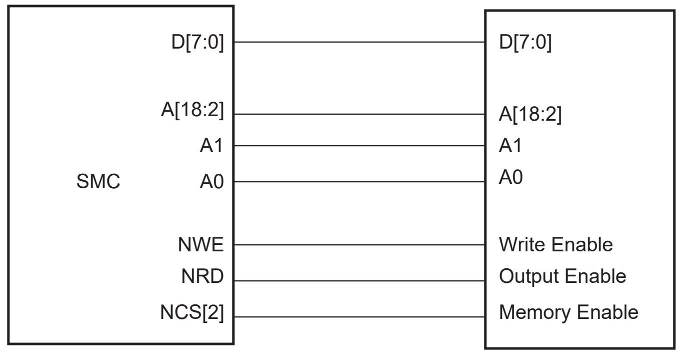
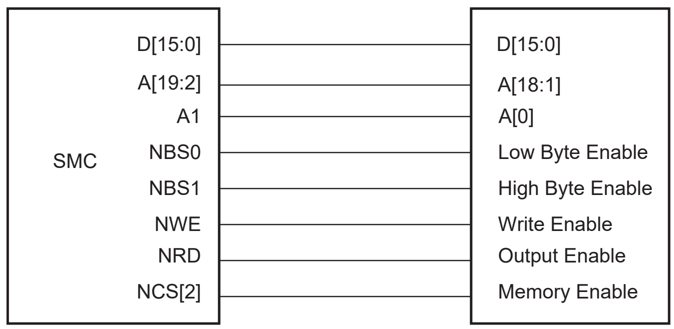


Figure 16-5. Memory Connection for a 16-bit Data Bus



16.9.2 Byte Write or Byte Select Access

Each chip select with a 16-bit data bus can operate with one of two different types of write access: Byte Write or Byte Select. This is controlled by HSMC_MODE.BAT for the corresponding chip select.

16.9.2.1 Byte Write Access

Byte Write Access is used to connect 2 x 8-bit devices as a 16-bit memory, and supports one write signal per byte of the data bus and a single read signal.

Note that the SMC does not allow boot in Byte Write Access mode.

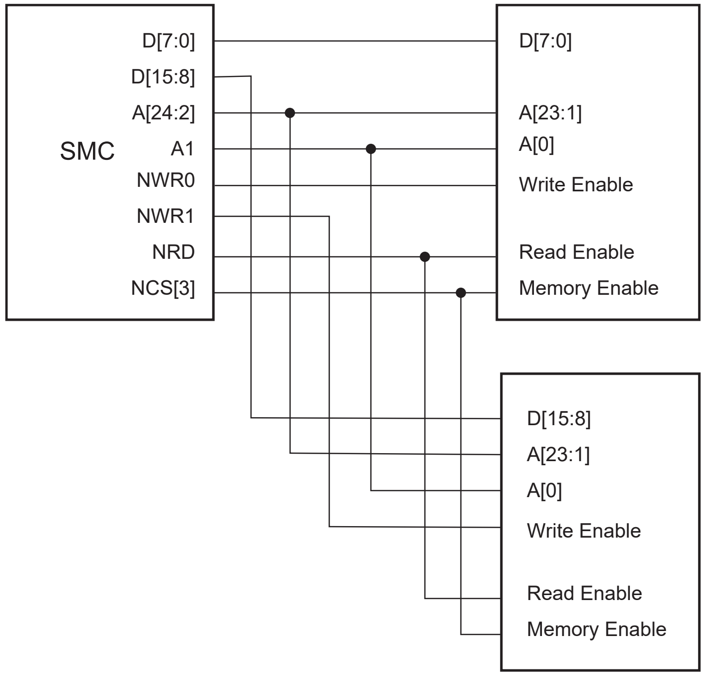
For 16-bit devices, the SMC provides NWR0 and NWR1 write signals for respectively Byte0 (lower byte) and Byte1 (upper byte) of a 16-bit bus. One single read signal (NRD) is provided.

16.9.2.2 Byte Select Access

Byte Select Access is used to connect one 16-bit device. In this mode, read/write operations can be enabled/disabled at Byte level. One Byte-select line per byte of the data bus is provided. One NRD and one NWE signal control read and write.

For 16-bit devices, the SMC provides NBS0 and NBS1 selection signals for respectively Byte0 (lower byte) and Byte1 (upper byte) of a 16-bit bus.

Figure 16-6. Connection of 2 x 8-bit Devices on a 16-bit Bus: Byte Write Option



16.9.2.3 Signal Multiplexing

Depending on the Byte Access Type (BAT), only the write signals or the byte select signals are used. To save IOs at the external bus interface, control signals at the SMC interface are multiplexed. The table below shows signal multiplexing depending on the data bus width and the Byte Access Type.

For 16-bit devices, bit A0 of the address is unused. When the Byte Select option is selected, NWR1 is unused. When the Byte Write option is selected, NBS0 is unused.

Table 16-3. SMC Multiplexed Signal Translation

Device Type	Signal Name		
	16-bit Bus		8-bit Bus
	1 x 16-bit	2 x 8-bit	1 x 8-bit
Byte Access Type (BAT)	Byte Select	Byte Write	–
NBS0_A0	NBS0	–	A0
NWE_NWR0	NWE	NWR0	NWE
NBS1_NWR1	NBS1	NWR1	–
A1	A1	A1	A1

16.10 Standard Read and Write Protocols

In the following sections, the Byte Access Type is not considered. Byte select lines (NBS0 to NBS1) always have the same timing as the A address bus. NWE represents either the NWE signal in byte select access type or one of the byte write lines (NWR0 to NWR1) in byte write access type. NWR0 to NWR1 have the same timings and protocol as NWE. In the same way, NCS represents one of the NCS[0..3] chip select lines.

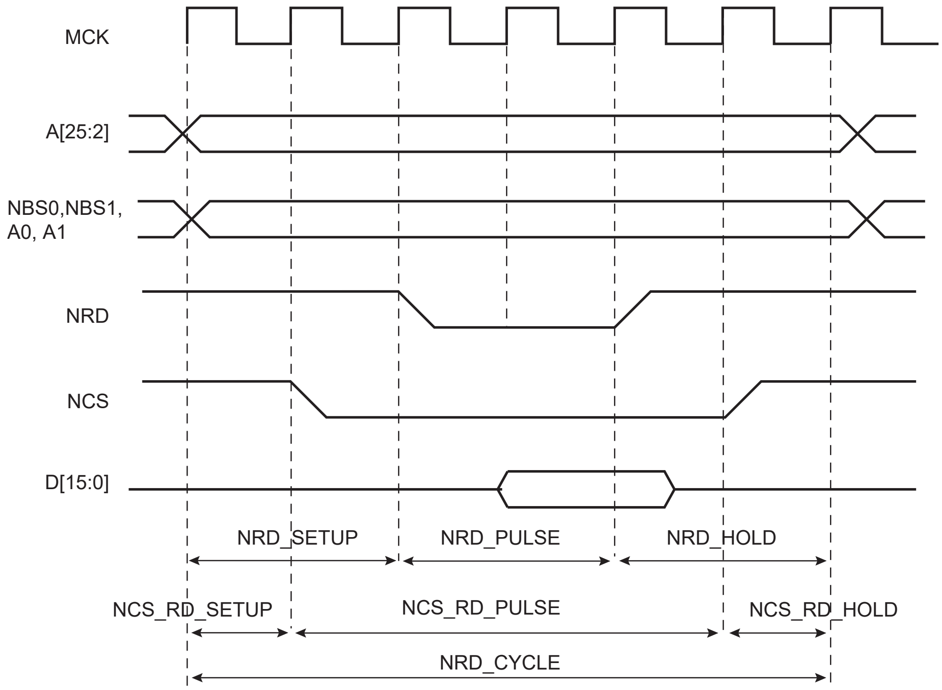
16.10.1 Read Waveforms

The read cycle is shown in the figure below.

The read cycle starts with the address setting on the memory address bus, i.e.:

- {A[25:2], A1, A0} for 8-bit devices
- {A[25:2], A1} for 16-bit devices

Figure 16-7. Standard Read Cycle



16.10.1.1 NRD Waveform

The NRD signal is characterized by a setup time, a pulse length and a hold time:

1. NRD_SETUP: the NRD setup time is defined as the setup of the address before the NRD falling edge.
2. NRD_PULSE: the NRD pulse length is the time between the NRD falling edge and the NRD rising edge.
3. NRD_HOLD: the NRD hold time is defined as the hold time of the address after the NRD rising edge.

16.10.1.2 NCS Waveform

Similar to the NRD signal, the NCS signal can be divided into a setup time, a pulse length and a hold time:

- NCS_RD_SETUP: the NCS setup time is defined as the setup time of the address before the NCS falling edge.
- NCS_RD_PULSE: the NCS pulse length is the time between the NCS falling edge and the NCS rising edge.
- NCS_RD_HOLD: the NCS hold time is defined as the hold time of the address after the NCS rising edge.

16.10.1.3 Read Cycle

The NRD_CYCLE time is defined as the total duration of the read cycle, that is, from the time when the address is set on the address bus to the point when the address may change. The total read cycle time is defined as:

- NRD_CYCLE = NRD_SETUP + NRD_PULSE + NRD_HOLD

as well as

- NRD_CYCLE = NCS_RD_SETUP + NCS_RD_PULSE + NCS_RD_HOLD

All NRD and NCS timings are defined separately for each chip select as an integer number of main system bus clock cycles. The NRD_CYCLE field is common to both the NRD and NCS signals, thus the timing period is of the same duration.

NRD_CYCLE, NRD_SETUP and NRD_PULSE implicitly define the NRD_HOLD value as:

- NRD_HOLD = NRD_CYCLE - NRD SETUP - NRD PULSE

NRD_CYCLE, NCS_RD_SETUP and NCS_RD_PULSE implicitly define the NCS_RD_HOLD value as:

- NCS_RD_HOLD = NRD_CYCLE - NCS_RD_SETUP - NCS_RD_PULSE

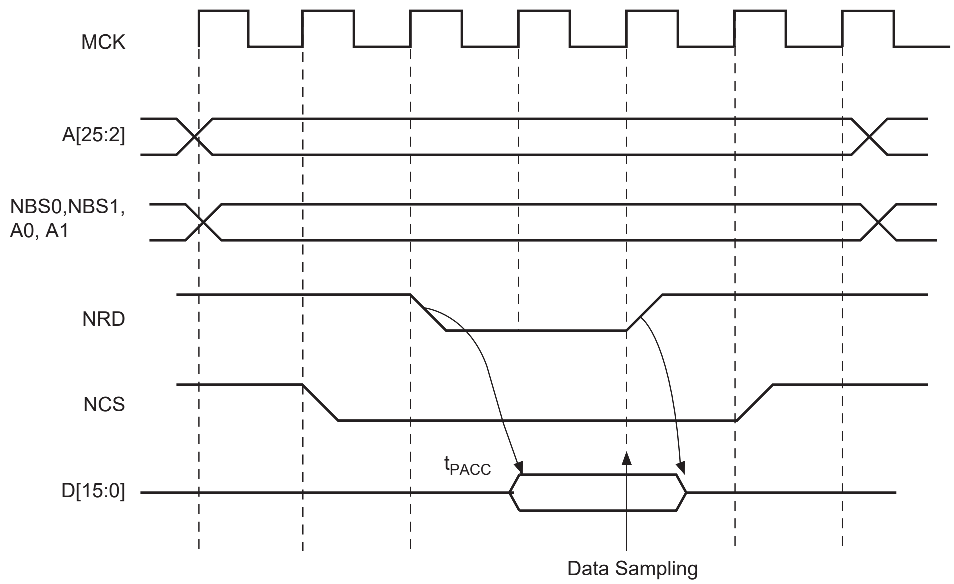
16.10.2 Read Mode

As NCS and NRD waveforms are defined independently of one another, the SMC needs to know when the read data is available on the data bus. The SMC does not compare NCS and NRD timings to know which signal rises first.
The READ_MODE parameter in the HSMC_MODE register of the corresponding chip select indicates which signal of NRD and NCS controls the read operation.

16.10.2.1 Read is Controlled by NRD (READ_MODE = 1)

The figure below shows the waveforms of a read operation of a typical asynchronous RAM. The read data is available t_{PACC} after the falling edge of NRD, and turns to ‘Z’ after the rising edge of NRD. In this case, the READ_MODE must be set to 1 (read is controlled by NRD), to indicate that data is available with the rising edge of NRD. The SMC samples the read data internally on the rising edge of the main system bus clock that generates the rising edge of NRD, whatever the programmed waveform of NCS.

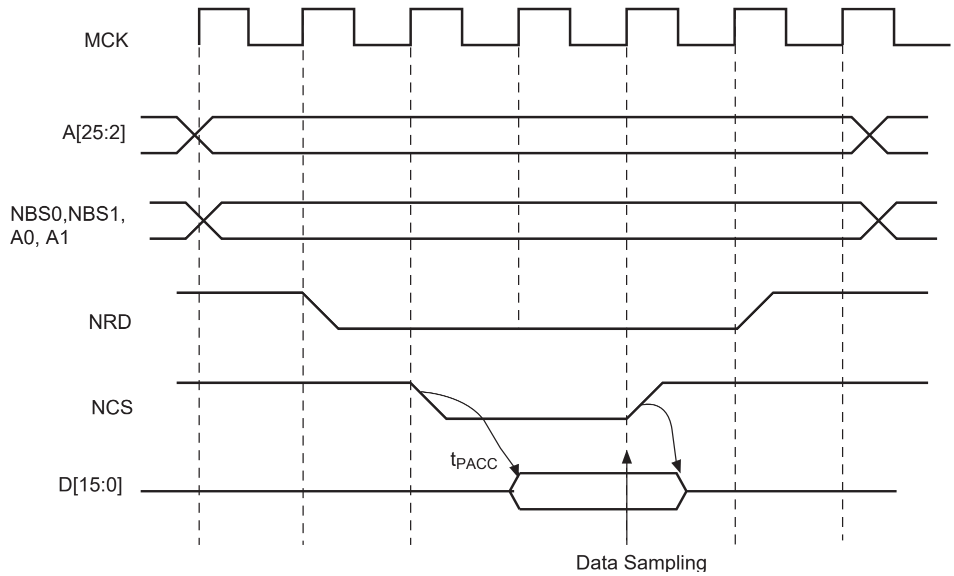
Figure 16-8. READ_MODE = 1: Data is Sampled by SMC before the Rising Edge of NRD



16.10.2.2 Read is Controlled by NCS (READ_MODE = 0)

The figure below shows the typical read cycle. The read data is valid t_{PACC} after the falling edge of the NCS signal and remains valid until the rising edge of NCS. Data must be sampled when NCS is raised. In that case, the READ_MODE must be configured to 0 (read is controlled by NCS): the SMC internally samples the data on the rising edge of the main system bus clock that generates the rising edge of NCS, whatever the programmed waveform of NRD.

Figure 16-9. READ_MODE = 0: Data is Sampled by SMC before the Rising Edge of NCS



16.10.3 Write Waveforms

The write protocol, similar to the read protocol, is depicted in the following figure. The write cycle starts with the address setting on the memory address bus.

16.10.3.1 NWE Waveforms

The NWE signal is characterized by a setup time, a pulse length and a hold time:

- NWE_SETUP: the NWE setup time is defined as the setup of the address and data before the NWE falling edge.
- NWE_PULSE: the NWE pulse length is the time between the NWE falling edge and the NWE rising edge.
- NWE_HOLD: the NWE hold time is defined as the hold time of the address and data after the NWE rising edge.

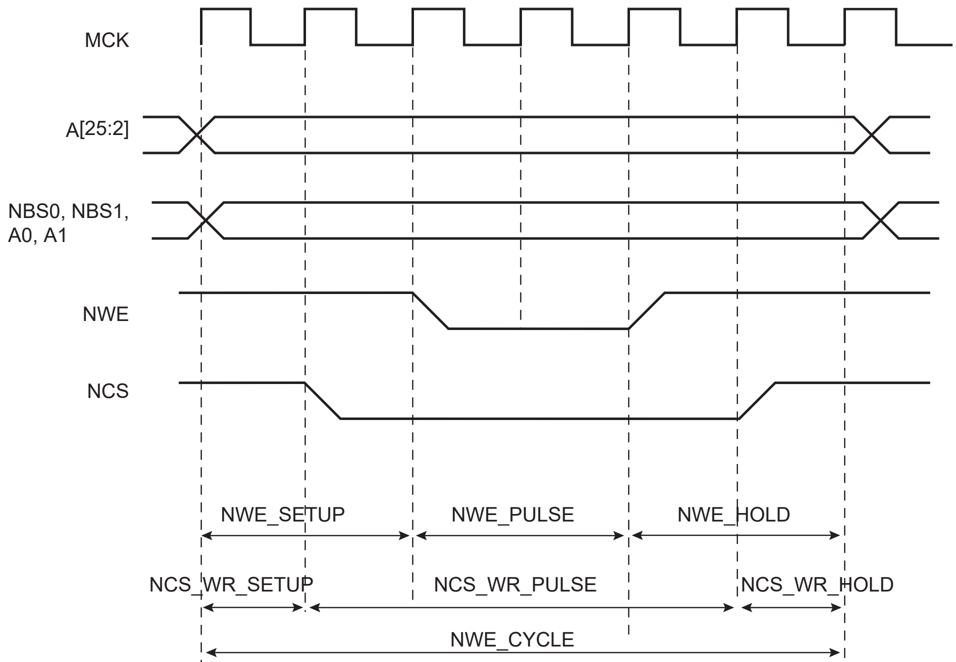
The NWE waveforms apply to all byte-write lines in Byte Write Access mode: NWR0 to NWR3.

16.10.3.2 NCS Waveforms

The NCS signal waveforms in write operations are not the same as those applied in read operations, but are defined separately:

- NCS_WR_SETUP: the NCS setup time is defined as the setup time of the address before the NCS falling edge.
- NCS_WR_PULSE: the NCS pulse length is the time between the NCS falling edge and the NCS rising edge.
- NCS_WR_HOLD: the NCS hold time is defined as the hold time of the address after the NCS rising edge.

Figure 16-10. Write Cycle



16.10.3.3 Write Cycle

The write cycle time is defined as the total duration of the write cycle, that is, from the time when the address is set on the address bus to the point when the address may change. The total write cycle time is equal to:

- NWE_CYCLE = NWE_SETUP + NWE_PULSE + NWE_HOLD

as well as

- NWE_CYCLE = NCS_WR_SETUP + NCS_WR_PULSE + NCS_WR_HOLD

All NWE and NCS (write) timings are defined separately for each chip select as an integer number of main system bus clock cycles. The NWE_CYCLE field is common to both the NWE and NCS signals, thus the timing period is of the same duration.

NWE_CYCLE, NWE_SETUP and NWE_PULSE implicitly define the NWE_HOLD value as:

- NWE_HOLD = NWE_CYCLE - NWE_SETUP - NWE_PULSE

NWE_CYCLE, NCS_WR_SETUP and NCS_WR_PULSE implicitly define the NCS_WR_HOLD value as:

- NCS_WR_HOLD = NWE_CYCLE - NCS_WR_SETUP - NCS_WR_PULSE

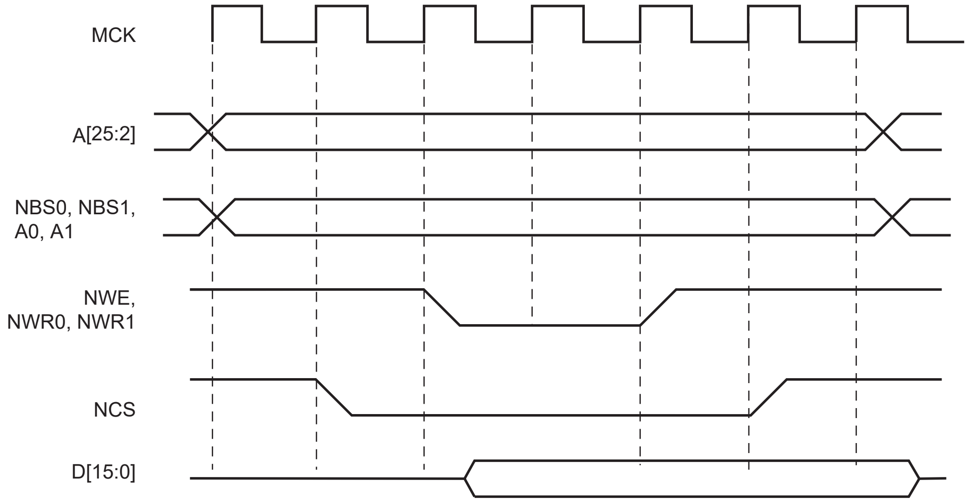
16.10.4 Write Mode

The WRITE_MODE parameter in the HSMC_MODE register of the corresponding chip select indicates which signal controls the write operation.

16.10.4.1 Write is Controlled by NWE (WRITE_MODE = 1)

The figure below shows the waveforms of a write operation with WRITE_MODE set to 1. The data is put on the bus during the pulse and hold steps of the NWE signal. The internal data buffers are switched to Output mode after the NWE_SETUP time, and until the end of the write cycle, regardless of the programmed waveform on NCS.

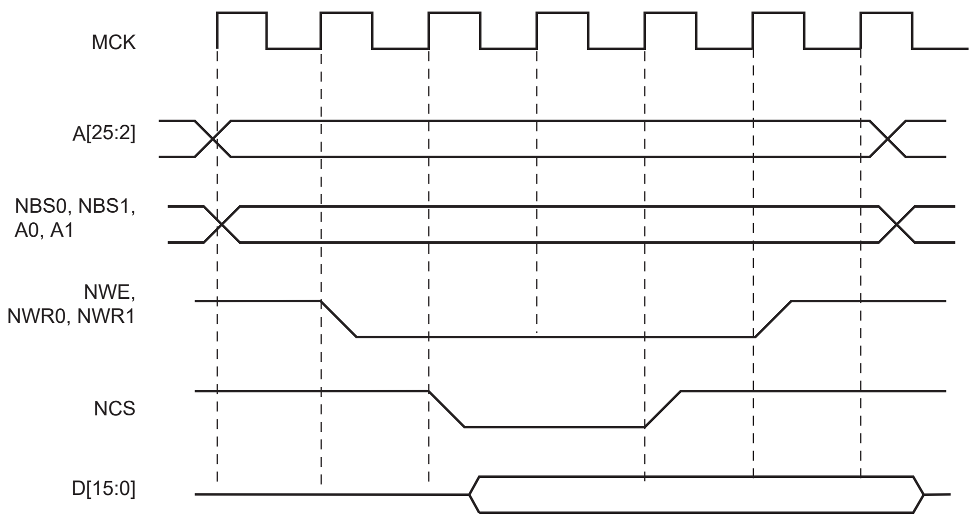
Figure 16-11. WRITE_MODE = 1. The write operation is controlled by NWE



16.10.4.2 Write is Controlled by NCS (WRITE_MODE = 0)

The figure below shows the waveforms of a write operation with WRITE_MODE configured to 0. The data is put on the bus during the pulse and hold steps of the NCS signal. The internal data buffers are switched to Output mode after the NCS_WR_SETUP time, and until the end of the write cycle, regardless of the programmed waveform on NWE.

Figure 16-12. WRITE_MODE = 0. The write operation is controlled by NCS

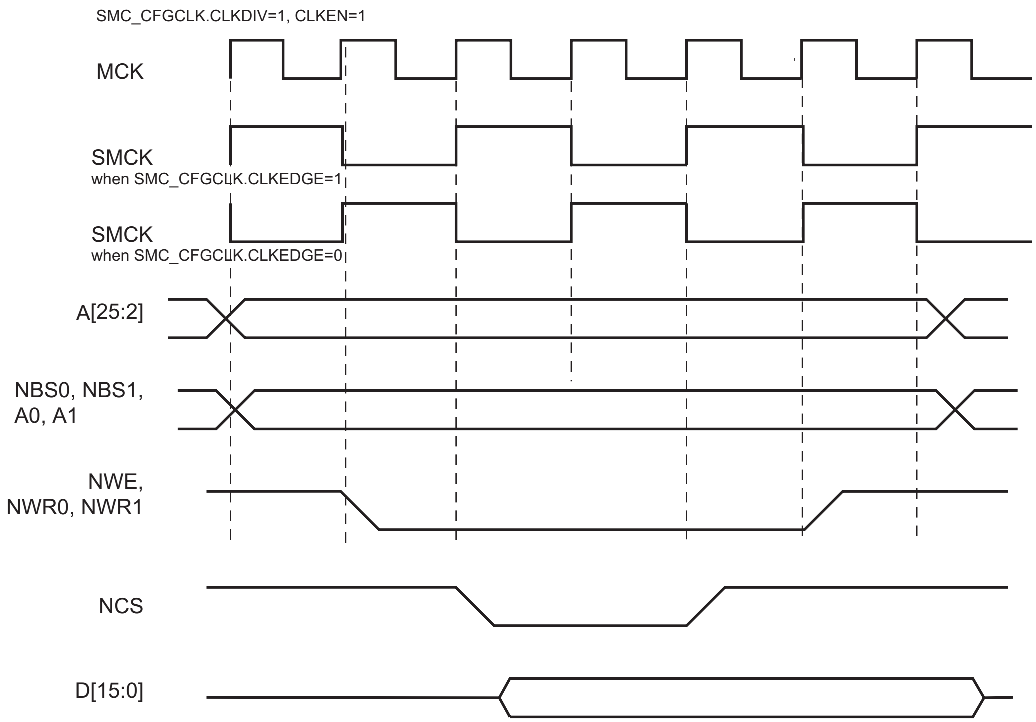


16.10.5 External Bus Clock

The SMC provides an external bus clock synchronous to the address, data and control signals. When SMC_CFGCLK.CLKEN=1, the frequency of the output clock is periph_clock/ (CLKDIV+1).

The start of the access to the external memory device can be aligned on the rising or falling edge of the output clock by configuring the SMC_CFGCLK.CLKEDGE.

Figure 16-13. SMC Clock Output Waveform



16.10.6 Coding Timing Parameters

All timing parameters are defined for one chip select and are grouped together in one register according to their type:

- The HSMC_SETUP register groups the definition of all setup parameters:
  NRD_SETUP, NCS_RD_SETUP, NWE_SETUP, NCS_WR_SETUP
- The HSMC_PULSE register groups the definition of all pulse parameters:
  NRD_PULSE, NCS_RD_PULSE, NWE_PULSE, NCS_WR_PULSE
- The HSMC_CYCLE register groups the definition of all cycle parameters:
  NRD_CYCLE, NWE_CYCLE

The table below shows how the timing parameters are coded and their permitted range.

Table 16-4. Coding and Range of Timing Parameters

Coded Value	Number of Bits	Effective Value	Permitted Range	
			Coded Value	Effective Value
setup [5:0]	6	128 x setup[5] + setup[4:0]	0 ≤ setup ≤ 31	0..31
			32 ≤ setup ≤ 63	128..(128 + 31)
pulse [6:0]	7	256 x pulse[6] + pulse[5:0]	0 ≤ pulse ≤ 63	0..63
			64 ≤ pulse ≤ 127	256..(256 + 63)
cycle[8:0]	9	256 x cycle[8:7] + cycle[6:0]	0 ≤ cycle ≤ 127	0..127
			128 ≤ cycle ≤ 255	256..(256 + 127)
			256 ≤ cycle ≤ 383	512..(512 + 127)
			384 ≤ cycle ≤ 511	768..(768 + 127)

16.10.7 Reset Values of Timing Parameters

The table below gives the default value of timing parameters at reset.

Table 16-5. Reset Values of Timing Parameters

Register	Reset Value	Description
HSMC_SETUP	0x01010101	All setup timings are set to 1.
HSMC_PULSE	0x01010101	All pulse timings are set to 1.
HSMC_CYCLE	0x00030003	The read and write operations last three main system bus clock cycles and provide one hold cycle.
WRITE_MODE	1	Write is controlled with NEW.
READ_MODE	1	Read is controlled with NRD.

16.10.8 Usage Restriction

The SMC does not check the validity of the user-programmed parameters. If the sum of SETUP and PULSE parameters is larger than the corresponding CYCLE parameter, this leads to an unpredictable behavior of the SMC.

16.10.8.1 For Read Operations

Null but positive setup and hold of address and NRD and/or NCS cannot be ensured at the memory interface because of the propagation delay of these signals through external logic and pads. When positive setup and hold values must be verified, then it is strictly recommended to program non-null values so as to cover possible skews between address, NCS and NRD signals.

16.10.8.2 For Write Operations

If a null hold value is programmed on NWE, the SMC can ensure a positive hold of address, byte select lines, and NCS signal after the rising edge of NWE. This is true for WRITE_MODE = 1 only. See Early Read Wait State.

16.10.8.3 For Read and Write Operations

A null value for pulse parameters is forbidden and may lead to an unpredictable behavior.

In read and write cycles, the setup and hold time parameters are defined in reference to the address bus. For external devices that require setup and hold time between NCS and NRD signals (read), or between NCS and NWE signals (write), these setup and hold times must be converted into setup and hold times in reference to the address bus.

16.11 Scrambling/Unscrambling Function

The external data bus D[15:0] can be scrambled in order to make recovery of intellectual property data located in off-chip memories more difficult by analyzing data at the package pin level of either the microcontroller or the memory device.

The scrambling and unscrambling are performed on-the-fly without additional wait states.

The scrambling method depends on two user-configurable key registers, HSMC_KEY1 and HSMC_KEY2. These key registers are only accessible in Write mode.

The key must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

The scrambling/unscrambling function is enabled or disabled by configuring specific bits in the HSMC_OCMS and the HSMC_TIMINGSx registers. The bit configuration values to enable memory scrambling are summarized in the table below.

Table 16-6. Scrambling Function Bit Encoding

Memories	Bit Values		
	HSMC_OCMS.SMSE	HSMC_OCMS.SRSE	HSMC_TIMINGSx.OCMS
Off-chip Memories	1	0	1
NAND Flash with NFC	0	1	0

When the NAND Flash memory content is scrambled, the on-chip NFC SRAM page buffer associated for the transfer is also scrambled.

16.12 Automatic Wait States

Under certain circumstances, the SMC automatically inserts idle cycles between accesses to avoid bus contention or operation conflict.

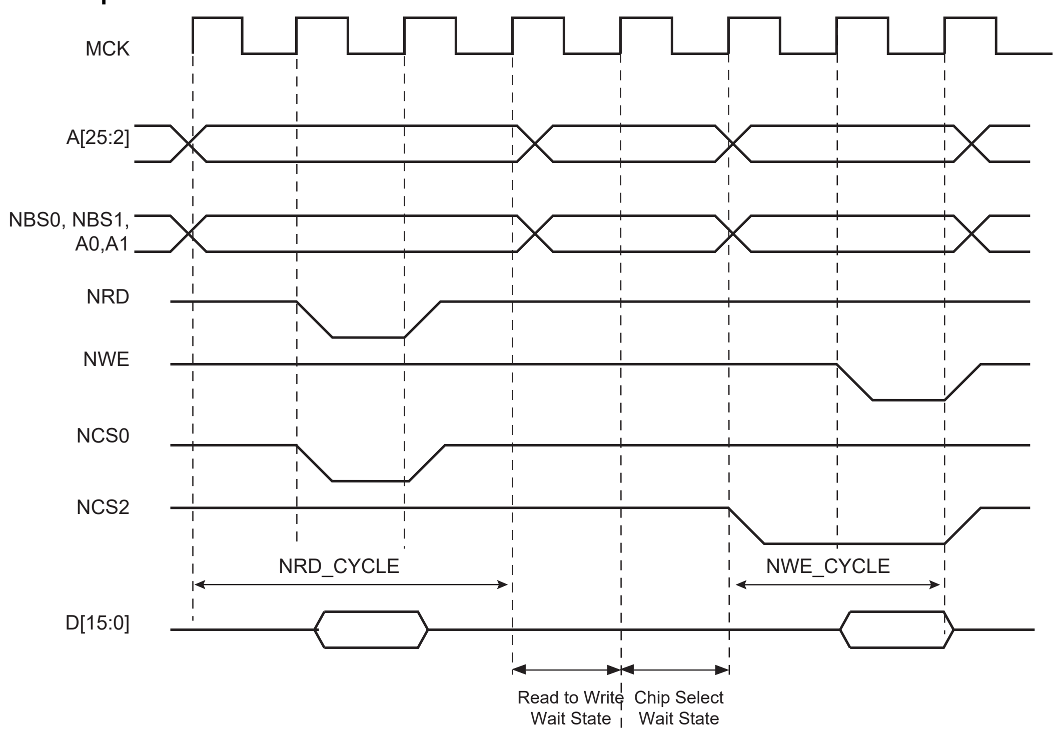
16.12.1 Chip Select Wait States

The SMC always inserts an idle cycle between two transfers on separate chip selects. This idle cycle ensures that there is no bus contention between the deactivation of one device and the activation of the next one.

During chip select wait state, all control lines are turned inactive: NBS0 to NBS1, NWR0 to NWR1, NCS[0..3], and NRD lines. They are all set to 1.

The figure below illustrates a chip select wait state between access on Chip Select 0 and Chip Select 2.

Figure 16-14. Chip Select Wait State between a Read Access on NCS0 and a Write Access on NCS2



16.12.2 Early Read Wait State

In some cases, the SMC inserts a wait state cycle between a write access and a read access to allow time for the write cycle to end before the subsequent read cycle begins. This wait state is not generated in addition to a chip select wait state. The early read cycle thus only occurs between a write and read access to the same memory device (same chip select).

An early read wait state is automatically inserted if at least one of the following conditions is valid:

- if the write controlling signal has no hold time and the read controlling signal has no setup time (see figure Early Read Wait State: Write with No Hold Followed by Read with No Setup).
- in NCS Write Controlled mode (WRITE_MODE = 0), if there is no hold timing on the NCS signal and the NCS_RD_SETUP parameter is configured to 0, regardless of the Read mode (see figure Early Read Wait State: NCS Controlled Write with No Hold Followed by a Read with No NCS Setup). The write operation must end with an NCS rising edge. Without an Early Read Wait State, the write operation could not complete properly.
- in NWE Controlled mode (WRITE_MODE = 1) and if there is no hold timing (NWE_HOLD = 0), the feedback of the write control signal is used to control address, data, chip select and byte select lines. If the external write control signal is not inactivated as expected due to load capacitances, an Early Read Wait State is inserted and address, data and control signals are maintained one more cycle. See figure Early Read Wait State: NWE-controlled Write with No Hold Followed by a Read with One Setup Cycle.

Figure 16-15. Early Read Wait State: Write with No Hold Followed by Read with No Setup

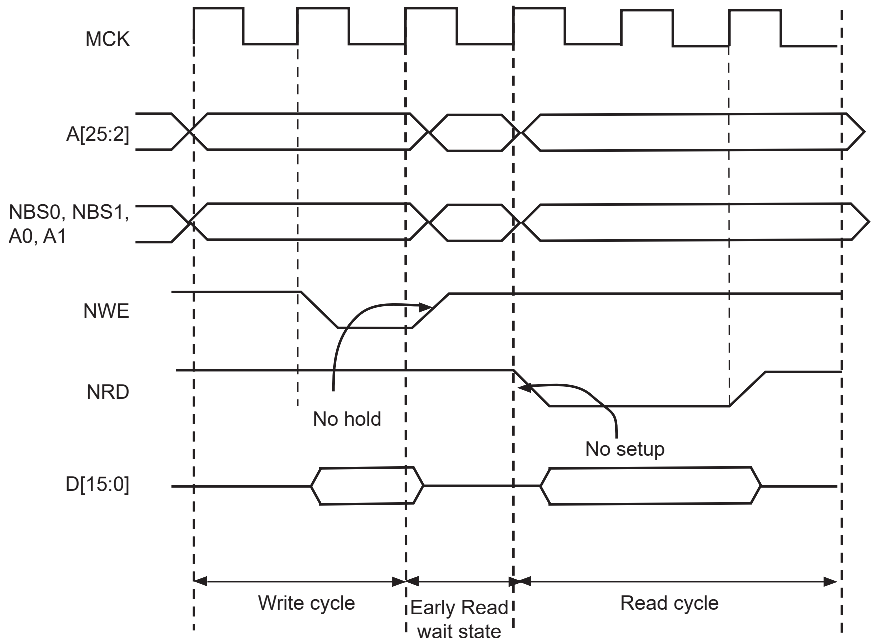


Figure 16-16. Early Read Wait State: NCS Controlled Write with No Hold Followed by a Read with No NCS Setup

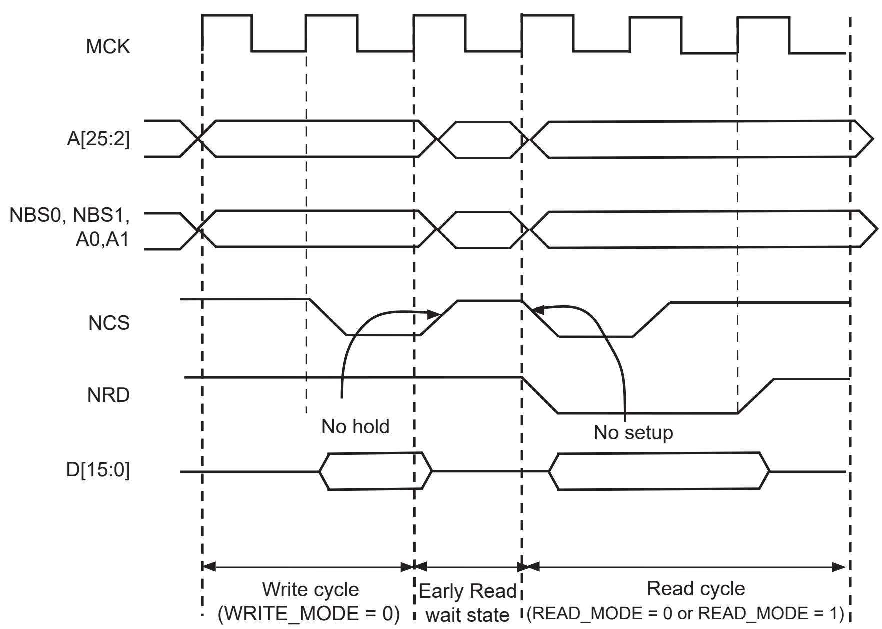
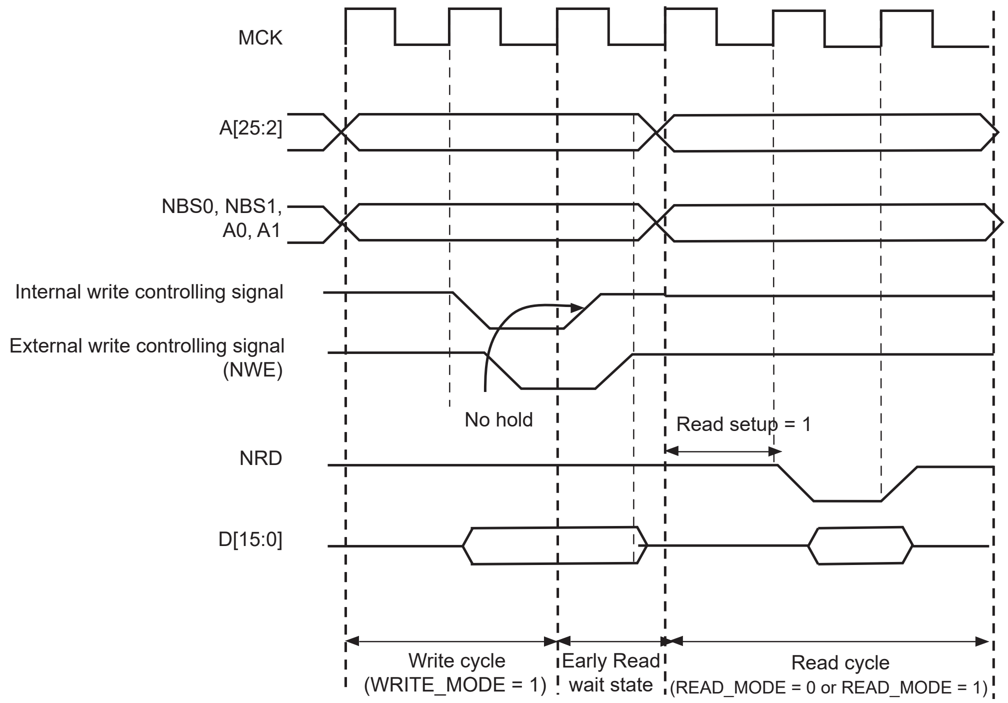


Figure 16-17. Early Read Wait State: NWE-controlled Write with No Hold Followed by a Read with One Setup Cycle



16.12.3 Reload User Configuration Wait State

The user may change any of the configuration parameters by writing the SMC user interface.

When detecting that a new user configuration has been written in the user interface, the SMC inserts a wait state before starting the next access. The so called “Reload User Configuration Wait State” is used by the SMC to load the new set of parameters to apply to next accesses.

The Reload Configuration Wait State is not applied in addition to the Chip Select Wait State. If accesses before and after reprogramming the user interface are made to different devices (Chip Selects), then one single Chip Select Wait State is applied.

On the other hand, if accesses before and after writing the user interface are made to the same device, a Reload Configuration Wait State is inserted, even if the change does not concern the current Chip Select.

16.12.3.1 User Procedure

To insert a Reload Configuration Wait State, the SMC detects a write access to any HSMC_MODE register of the user interface. If only the timing registers are modified (HSMC_SETUP, HSMC_PULSE, HSMC_CYCLE registers) in the user interface, the user must validate the modification by writing the HSMC_MODE register, even if no change was made on the mode parameters.

16.12.3.2 Slow Clock Mode Transition

A Reload Configuration Wait state is also inserted when Slow Clock mode is entered or exited, after the end of the current transfer (see Slow Clock Mode).

16.12.4 Read to Write Wait State

Due to an internal mechanism, a wait cycle is always inserted between consecutive read and write SMC accesses.

This wait cycle is referred to as a read to write wait state in this document.

This wait cycle is applied in addition to chip select and reload user configuration wait states when they are to be inserted. See figure Chip Select Wait State between a Read Access on NCS0 and a Write Access on NCS2.

16.13 External Wait

Any access can be extended by an external device using the NWAIT input signal of the SMC. The EXNW_MODE field of the HSMC_MODE register on the corresponding chip select must be set to either ‘10’ (Frozen mode) or ‘11’ (Ready mode). When the EXNW_MODE is set to ‘00’ (disabled), the NWAIT signal is simply ignored on the corresponding chip select. The NWAIT signal delays the read or write operation related to the read or write controlling signal, depending on the Read and Write modes of the corresponding chip select.

16.13.1 Restriction

When one of the EXNW_MODE is enabled, at least one hold cycle must be programmed for the read/write controlling signal. For that reason, the NWAIT signal cannot be used in Slow Clock mode (see Slow Clock Mode).

The NWAIT signal is assumed to be a response of the external device to the read/write request of the SMC. NWAIT is then examined by the SMC in the Pulse state of the read or write controlling signal. The assertion of the NWAIT signal outside the expected period has no impact on the SMC behavior.

16.13.2 Frozen Mode

When the external device asserts the NWAIT signal (active low), and after an internal synchronization of this signal, the SMC state is frozen, i.e., SMC internal counters are frozen, and all control signals remain unchanged. When the resynchronized NWAIT signal is deasserted, the SMC completes the access, resuming the access from the point where it was stopped. See the figure below. This mode must be selected when the external device uses the NWAIT signal to delay the access and to freeze the SMC.

The assertion of the NWAIT signal outside the expected period is ignored as illustrated in the figure Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10).

Figure 16-18. Write Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)

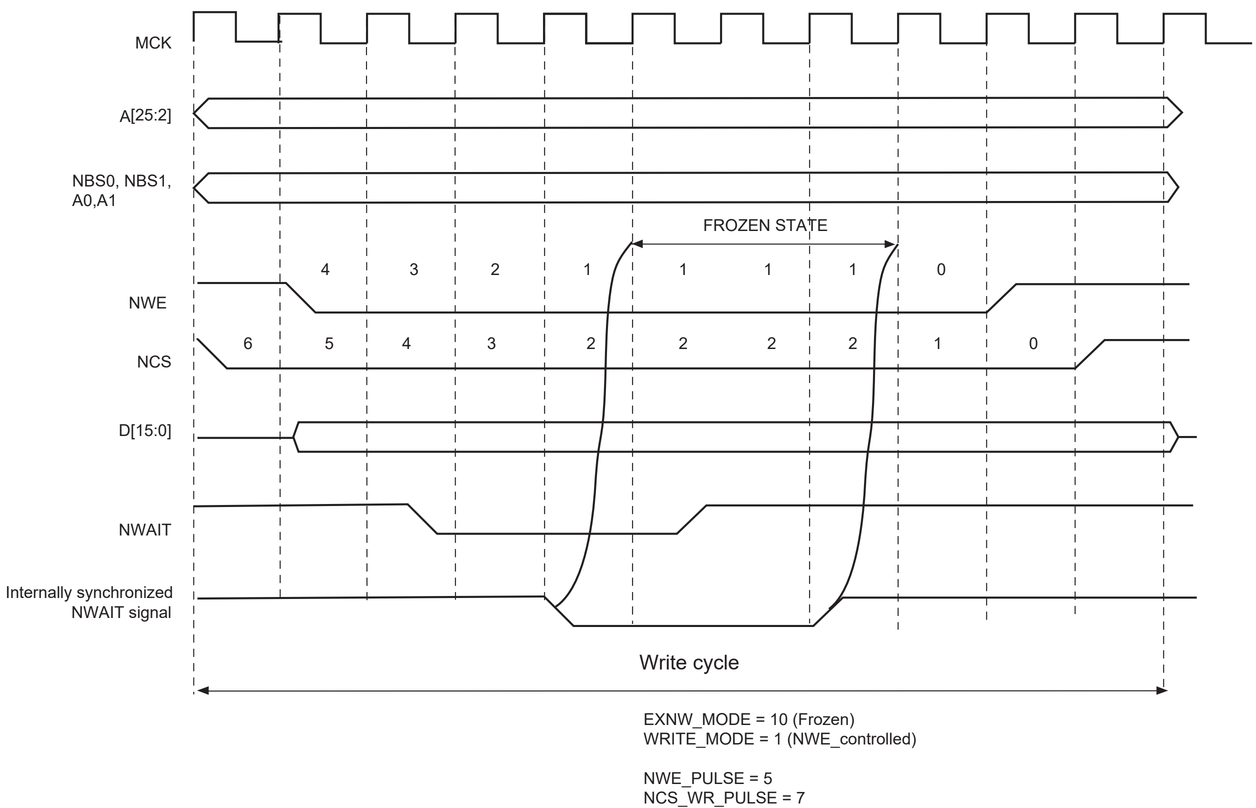
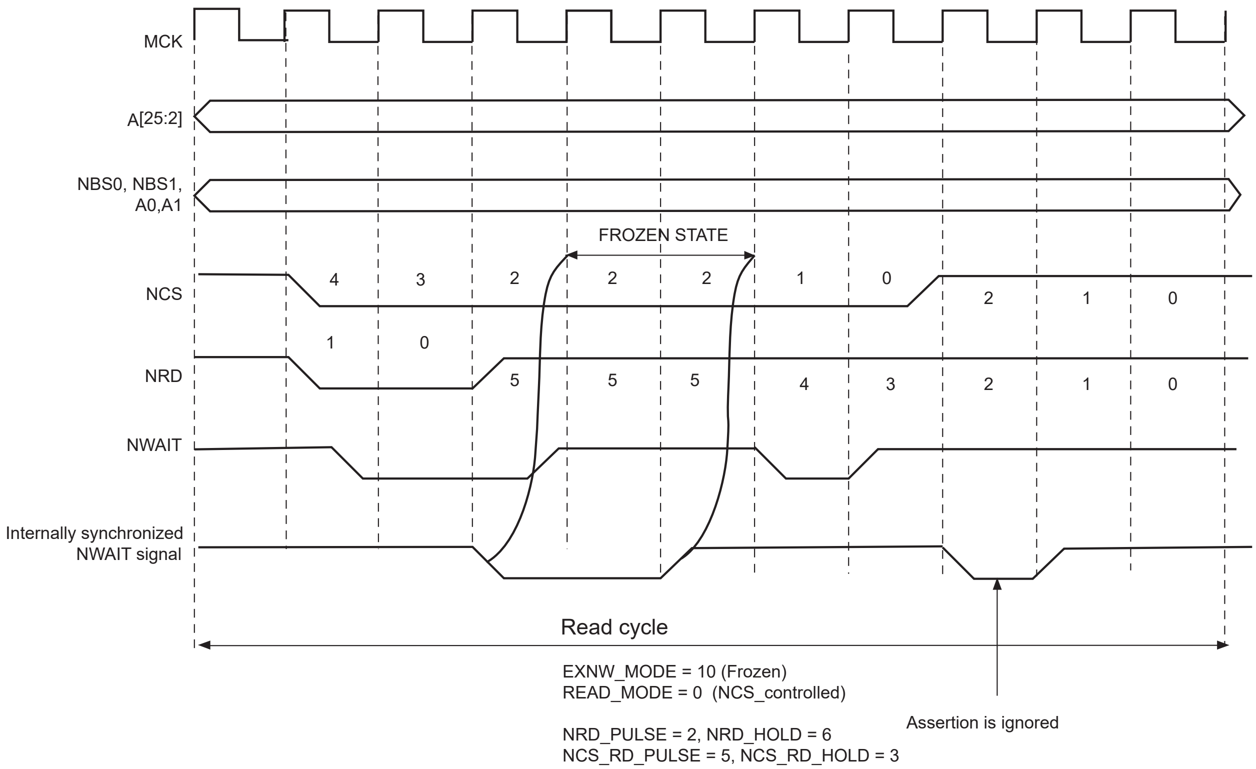


Figure 16-19. Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



16.13.3 Ready Mode

In Ready mode (EXNW_MODE = 11), the SMC behaves differently. Normally, the SMC begins the access by down counting the setup and pulse counters of the read/write controlling signal. In the last cycle of the pulse phase, the resynchronized NWAIT signal is examined.

If asserted, the SMC suspends the access as shown in the figures below. After deassertion, the access is completed: the hold step of the access is performed.

This mode must be selected when the external device uses deassertion of the NWAIT signal to indicate its ability to complete the read or write operation.

If the NWAIT signal is deasserted before the end of the pulse, or asserted after the end of the pulse of the controlling read/write signal, it has no impact on the access length as shown in the figure NWAIT Assertion in Read Access: Ready Mode (EXNW_MODE = 11).

Figure 16-20. NWAIT Assertion in Write Access: Ready Mode (EXNW_MODE = 11)

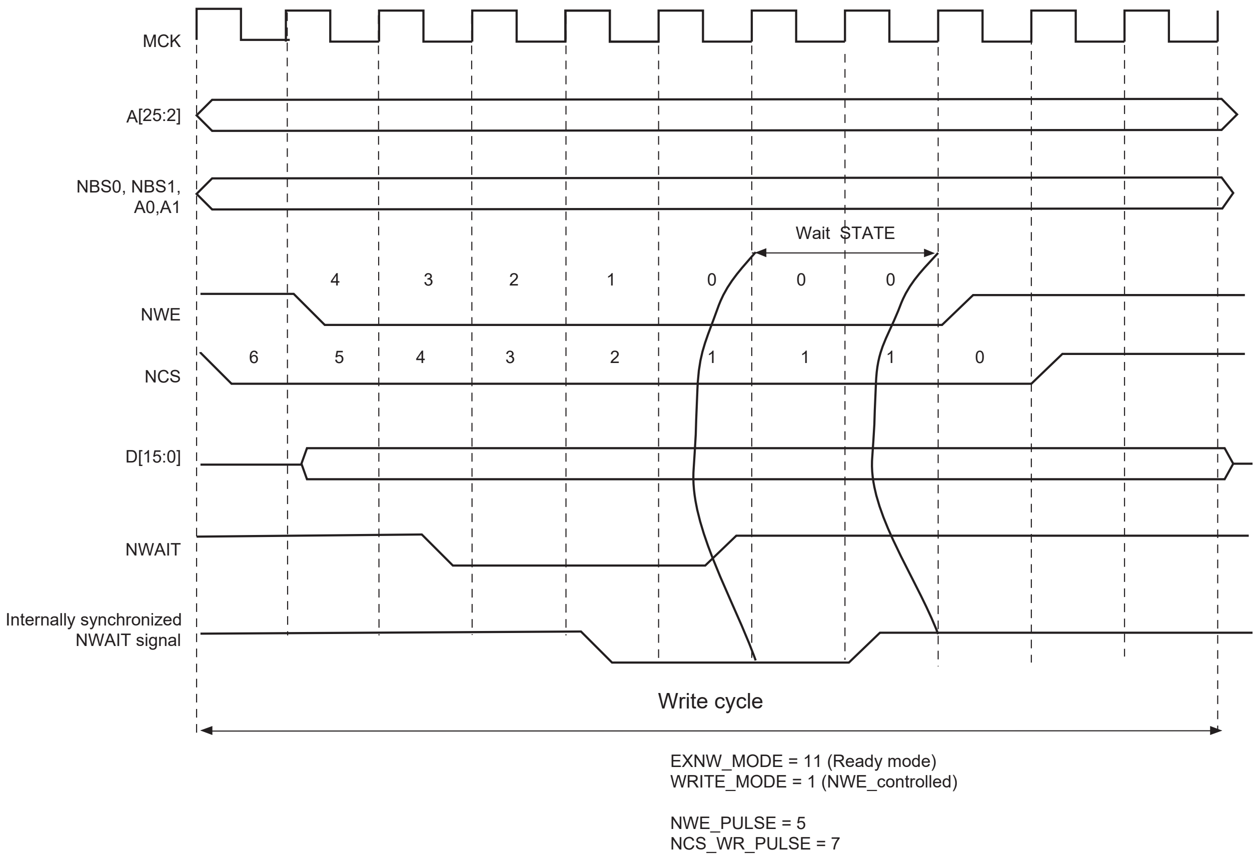
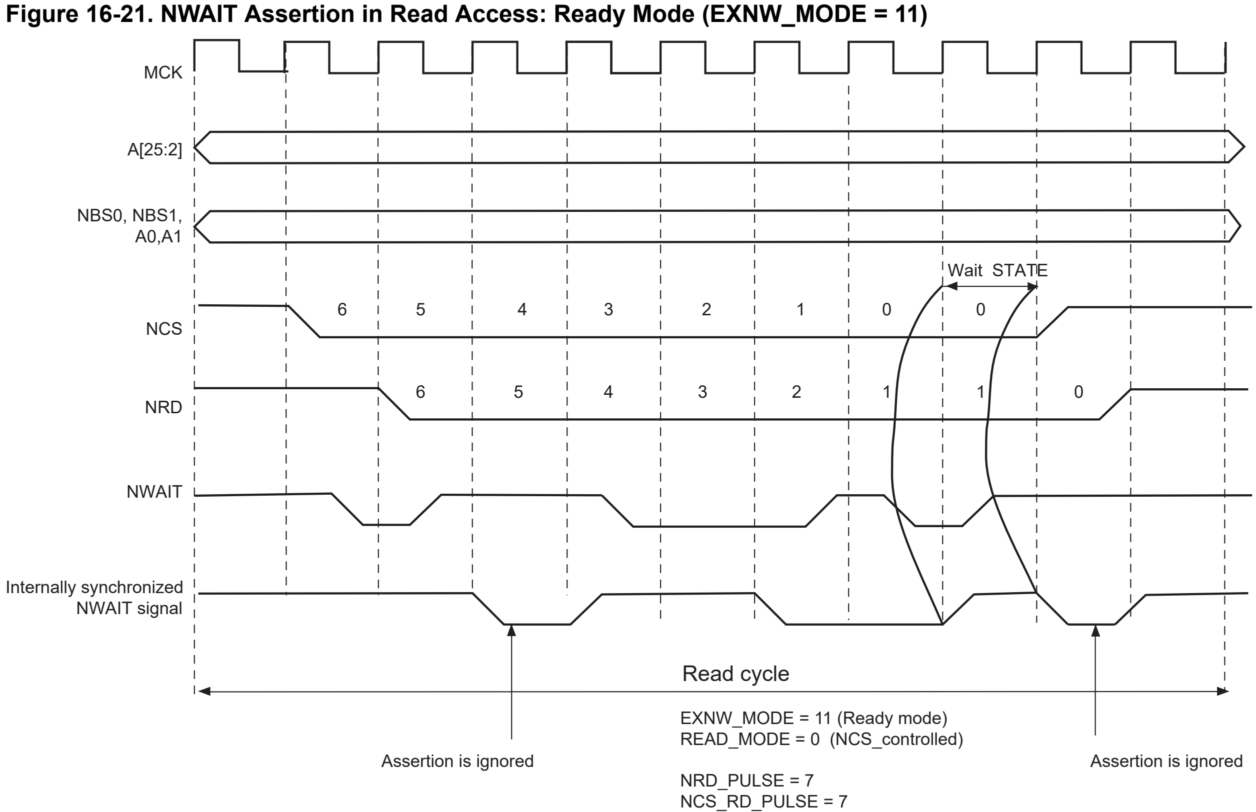


Figure 16-21. NWAIT Assertion in Read Access: Ready Mode (EXNW_MODE = 11)



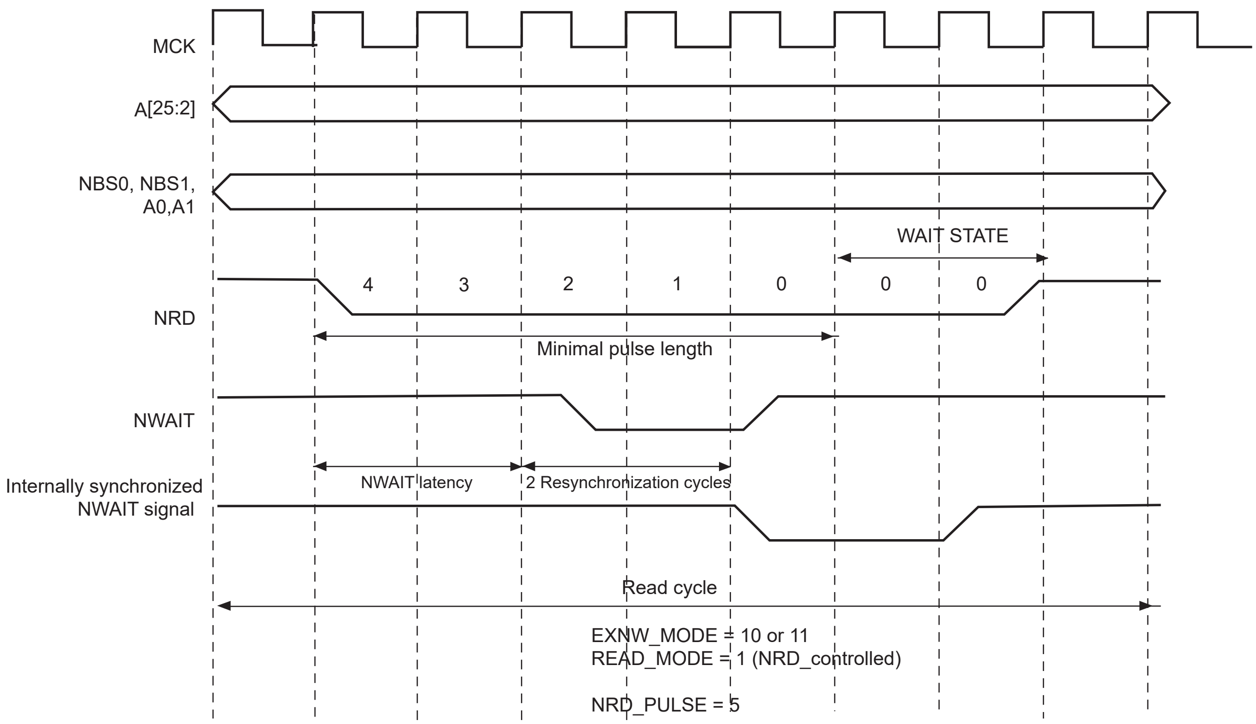
16.13.4 NWAIT Latency and Read/Write Timings

There may be a latency between the assertion of the read/write controlling signal and the assertion of the NWAIT signal by the device. The programmed pulse length of the read/write controlling signal must be at least equal to this latency plus the 2 cycles of resynchronization + 1 cycle. Otherwise, the SMC may enter the hold state of the access without detecting the NWAIT signal assertion. This is true in Frozen mode as well as in Ready mode. This is illustrated in the figure below.

When EXNW_MODE is enabled (ready or frozen), the user must program a pulse length of the read and write controlling signal of at least:

minimal pulse length = NWAIT latency + 2 resynchronization cycles + 1 cycle

Figure 16-22. NWAIT Latency



16.14 Slow Clock Mode

The SMC is able to automatically apply a set of “Slow Clock mode” read/write waveforms when an internal signal driven by the Power Management Controller is asserted because MCK has been turned to a very slow clock rate (typically 32 kHz clock rate). In this mode, the user-programmed waveforms are ignored and the Slow Clock mode waveforms are applied. This mode is provided so as to avoid reprogramming the User Interface with appropriate waveforms at very slow clock rate. When activated, the Slow mode is active on all chip selects.

16.14.1 Slow Clock Mode Waveforms

The figure below illustrates the read and write operations in Slow Clock mode. They are valid on all chip selects. The table below indicates the value of read and write parameters in Slow Clock mode.

Figure 16-23. Write/Read Cycles in Slow Clock Mode

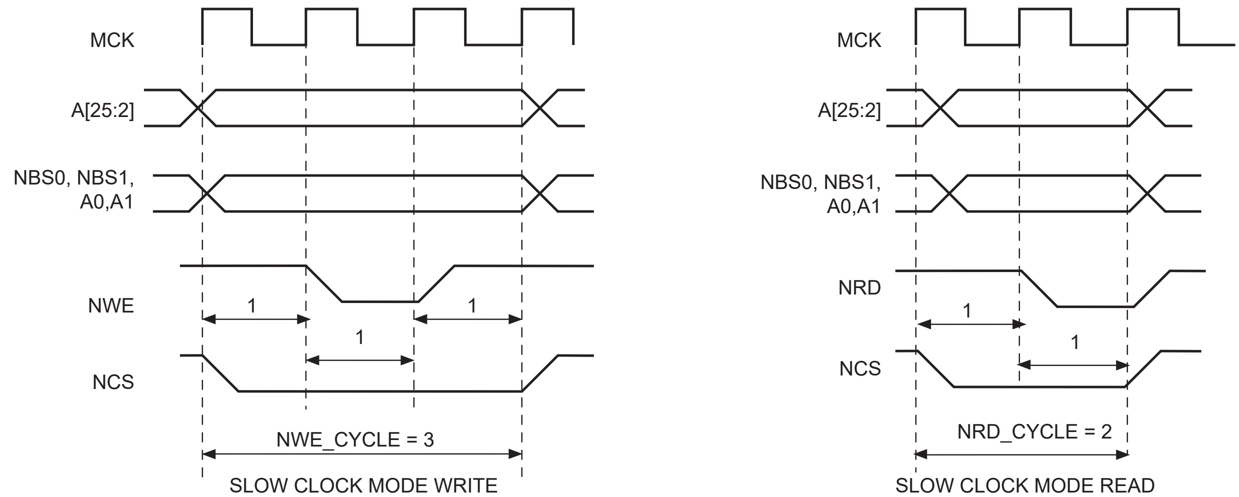


Table 16-7. Read and Write Timing Parameters in Slow Clock Mode

Read Parameters	Duration (cycles)	Write Parameters	Duration (cycles)
NRD_SETUP	1	NWE_SETUP	1
NRD_PULSE	1	NWE_PULSE	1
NCS_RD_SETUP	0	NCS_WR_SETUP	0
NCS_RD_PULSE	2	NCS_WR_PULSE	3
NRD_CYCLE	2	NWE_CYCLE	3

16.14.2 Switching from (to) Slow Clock Mode to (from) Normal Mode

When switching from Slow Clock mode to Normal mode, the current Slow Clock mode transfer is completed at high clock rate, with the set of Slow Clock mode parameters. See the figure Clock Rate Transition occurs while the SMC is performing a Write Operation. The external device may not be fast enough to support such timings.

The figure Recommended Procedure to Switch from Slow Clock Mode to Normal Mode or from Normal Mode to Slow Clock Mode illustrates the recommended procedure to properly switch from one mode to the other.

Figure 16-24. Clock Rate Transition occurs while the SMC is performing a Write Operation

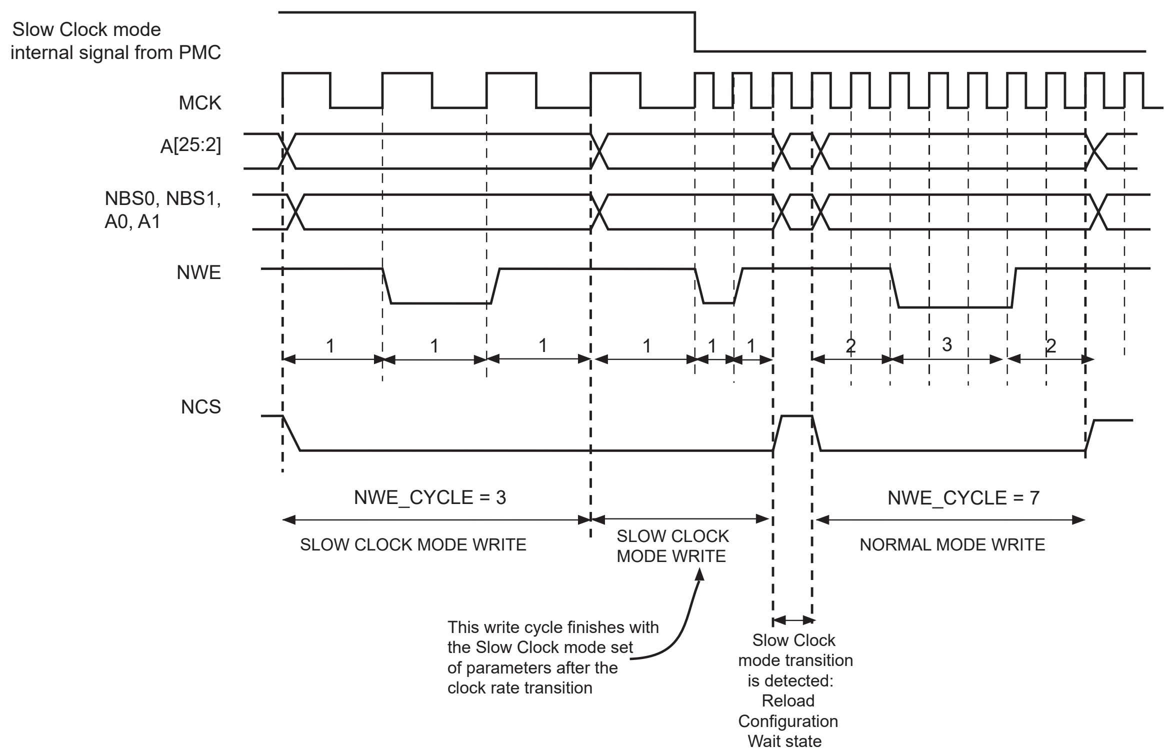
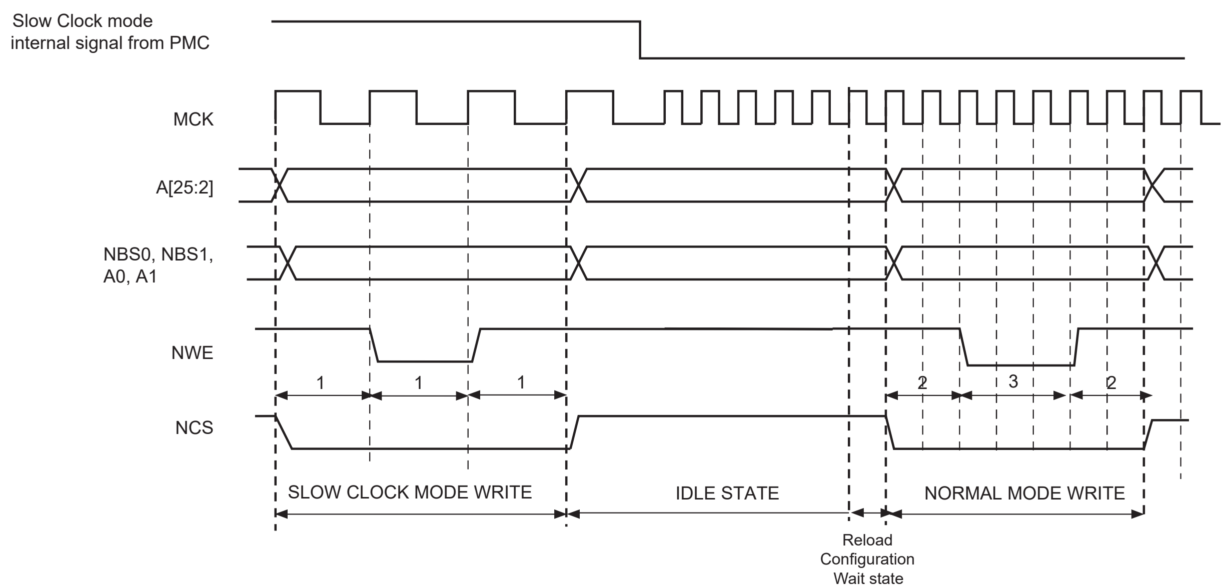


Figure 16-25. Recommended Procedure to Switch from Slow Clock Mode to Normal Mode or from Normal Mode to Slow Clock Mode



16.15 Register Write Protection

To prevent any single software error that may corrupt SMC behavior, selected registers can be write-protected by setting HSMC_WPMR.WPEN.

If a write access in a write-protected register is detected, then the WPVS flag in the Write Protection Status Register (HSMC_WPSR) is set and the field WPVSRC indicates in which register the write access has been attempted.

The WPVS flag is automatically reset after reading HSMC_WPSR.

The following registers can be write-protected:

- Setup Register
- Pulse Register
- Cycle Register
- Timings Register
- Mode Register

16.16 NFC Operations

16.16.1 NFC Overview

The NFC handles all the command, address and data sequences of the NAND low level protocol. An SRAM is used as an internal read/write buffer when data is transferred from or to the NAND.

16.16.2 NFC Control Registers

NAND Flash Read and NAND Flash Program operations can be performed through the NFC Command registers. In order to minimize CPU intervention and latency, commands are posted in a command buffer. This buffer provides zero wait state latency.

The NFC handles an automatic transfer between the external NAND Flash and the chip via the NFC SRAM. The transfer is done by programming NFC Command registers.

NFC Command registers are very efficient. When writing to these registers:

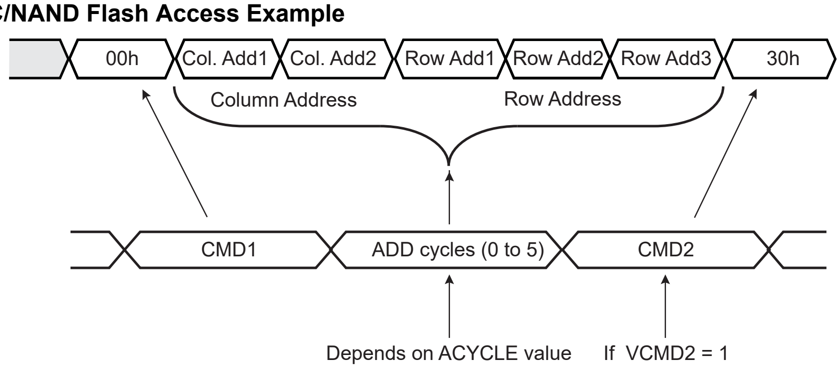
- the address of the register (NFCADDR_CMD) is the command used,
- the data of the register (NFCDATA_ADDT) is the address to be sent to the NAND Flash.

In one single access, the command is sent and immediately executed by the NFC. Two commands can even be programmed within a single access (CMD1, CMD2) depending on the VCMD2 value.

The NFC can send up to five address cycles.

The figure below shows a typical NAND Flash Page Read Command of a NAND Flash Memory and correspondence with the NFC Address Command register.

Figure 16-26. NFC/NAND Flash Access Example



For more details, see NFCADDR_CMD.

Reading the NFC Command register (to any address) gives the status of the NFC. This is especially useful to know if the NFC is busy, for example.

16.16.2.1 Building NFC Address Command Example

The base address is the HOST_ADDR address.

Page read operation example:

```
// Build the Address Command (NFCADDR_CMD)
AddressCommand = (HOST_ADDR |
  NFCWR=0    | // NFC Read Data from NAND Flash
  DATAEN=1   | // NFC Data phase Enable.
  CSID=1     | // Chip Select ID = 1
  ACYCLE= 5  | // Number of address cycle.
  VCMD2=1    | // CMD2 is sent after Address Cycles
  CMD2=0x30  | // CMD2 = 30h
  CMD1=0x0) // CMD1 = Read Command = 00h
 // Set the Address for Cycle 0
 HSMC_ADDR = Col. Add1
 // Write command with the Address Command built above
*AddressCommand = (Col. Add2 | // ADDR_CYCLE1
  Row Add1   | // ADDR_CYCLE2
  Row Add2   | // ADDR_CYCLE3
  Row Add3 )   // ADDR_CYCLE4
```

16.16.2.2 NFC Address Command

Name: NFCADDR_CMD
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						NFCWR	DATAEN	CSID[2]
Access						R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	CSID[1:0]		ACYCLE[2:0]			VCMD2	CMD2[7:6]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	CMD2[5:0]						CMD1[7:6]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	CMD1[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset								

Bit 26 – NFCWR NFC Write Enable

Value	Description
0	NFC reads data from the NAND Flash.
1	NFC writes data into the NAND Flash.

Bit 25 – DATAEN NFC Data Phase Enable
When set to true, the NFC will automatically read or write data after the command.

Bits 24:22 – CSID[2:0] Chip Select Identifier
Chip select used.

Bits 21:19 – ACYCLE[2:0] Number of Address Cycles Required for the Current Command
When ACYCLE field is different from zero, ACYCLE Address cycles are performed after Command Cycle 1. The maximum number of cycles is 5.

Bit 18 – VCMD2 Valid Cycle 2 Command
When set to true, the CMD2 field is issued after the address cycle.

Bits 17:10 – CMD2[7:0] Command Register Value for Cycle 2
When a write access occurs with the VCMD2 field set, the NFC sends this command after CMD1.

Bits 9:2 – CMD1[7:0] Command Register Value for Cycle 1
When a write access occurs, the NFC sends this command.

16.16.2.3 NFC Data Address

Name: NFCDATA_ADDT
Property: Write-only

If five address cycles are used, the first address cycle is ADDR_CYCLE0. See HSMC_ADDR.

Bit	31	30	29	28	27	26	25	24
	ADDR_CYCLE4[7:0]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDR_CYCLE3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR_CYCLE2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR_CYCLE1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset								

Bits 31:24 – ADDR_CYCLE4[7:0] NAND Flash Array Address Cycle 4
When less than five address cycles are used, ADDR_CYCLE4 is the fourth byte written to the NAND Flash.
When five address cycles are used, ADDR_CYCLE4 is the fifth byte written to NAND Flash.

Bits 23:16 – ADDR_CYCLE3[7:0] NAND Flash Array Address Cycle 3
When less than five address cycles are used, ADDR_CYCLE3 is the third byte written to the NAND Flash.
When five address cycles are used, ADDR_CYCLE3 is the fourth byte written to NAND Flash.

Bits 15:8 – ADDR_CYCLE2[7:0] NAND Flash Array Address Cycle 2
When less than five address cycles are used, ADDR_CYCLE2 is the second byte written to the NAND Flash.
When five address cycles are used, ADDR_CYCLE2 is the third byte written to NAND Flash.

Bits 7:0 – ADDR_CYCLE1[7:0] NAND Flash Array Address Cycle 1
When less than five address cycles are used, ADDR_CYCLE1 is the first byte written to the NAND Flash.
When five address cycles are used, ADDR_CYCLE1 is the second byte written to NAND Flash.

16.16.2.4 NFC Data Status

Name: NFCDATA_STATUS
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					NFCBUSY	NFCWR	DATAEN	CSID[2]
Access					R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	CSID[1:0]		ACYCLE[2:0]			VCMD2	CMD2[7:6]	
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	CMD2[5:0]						CMD1[7:6]	
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	CMD1[5:0]							
Access	R	R	R	R	R	R		
Reset								

Bit 27 – NFCBUSY NFC Busy Status Flag
If set to true, it indicates that the NFC is busy.

Bit 26 – NFCWR NFC Write Enable

Value	Description
0	NFC is in Read mode.
1	NFC is in Write mode.

Bit 25 – DATAEN NFC Data Phase Enable
When set to true, the NFC data phase is enabled.

Bits 24:22 – CSID[2:0] Chip Select Identifier
Chip select used.

Bits 21:19 – ACYCLE[2:0] Number of Address Cycles Required for the Current Command
When ACYCLE is different from zero, ACYCLE Address cycles are performed after Command Cycle 1.

Bit 18 – VCMD2 Valid Cycle 2 Command
When set to true, the CMD2 field is issued after the address cycle.

Bits 17:10 – CMD2[7:0] Command Register Value for Cycle 2
When VCMD2 bit is set to true, the Physical Memory Interface drives the IO bus with CMD2 field during the Command Latch cycle 2.

Bits 9:2 – CMD1[7:0] Command Register Value for Cycle 1
When a Read or Write Access occurs, the Physical Memory Interface drives the IO bus with CMD1 field during the Command Latch cycle 1.

16.16.3 NFC Initialization

Prior to any Command and Data Transfer, the SMC User Interface must be configured to meet the device timing requirements.

- Write Enable Configuration

Use NWE_SETUP, NWE_PULSE and NWE_CYCLE to define the write enable waveform according to the external device datasheet.

Use HSMC_TIMINGS.TADL to configure the timing between the last address latch cycle and the first rising edge of WEN for data input.

Figure 16-27. Write Enable Timing Configuration

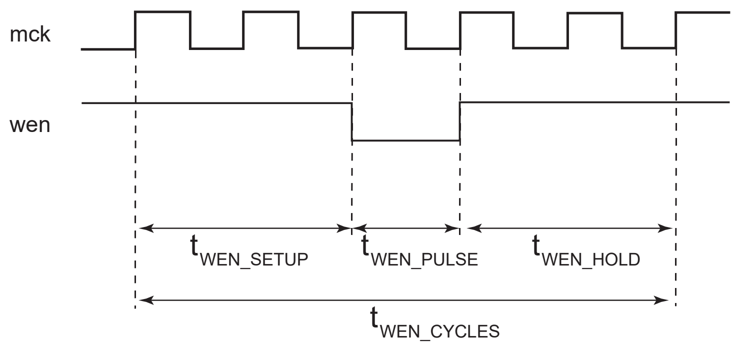
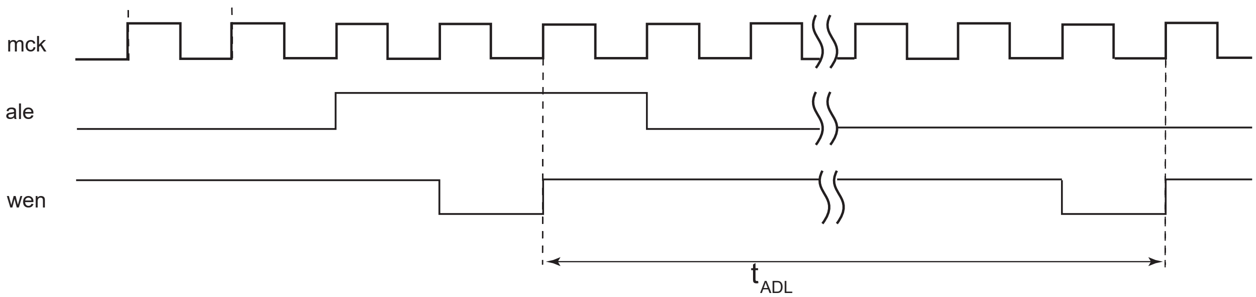


Figure 16-28. Write Enable Timing for NAND Flash Device Data Input Mode



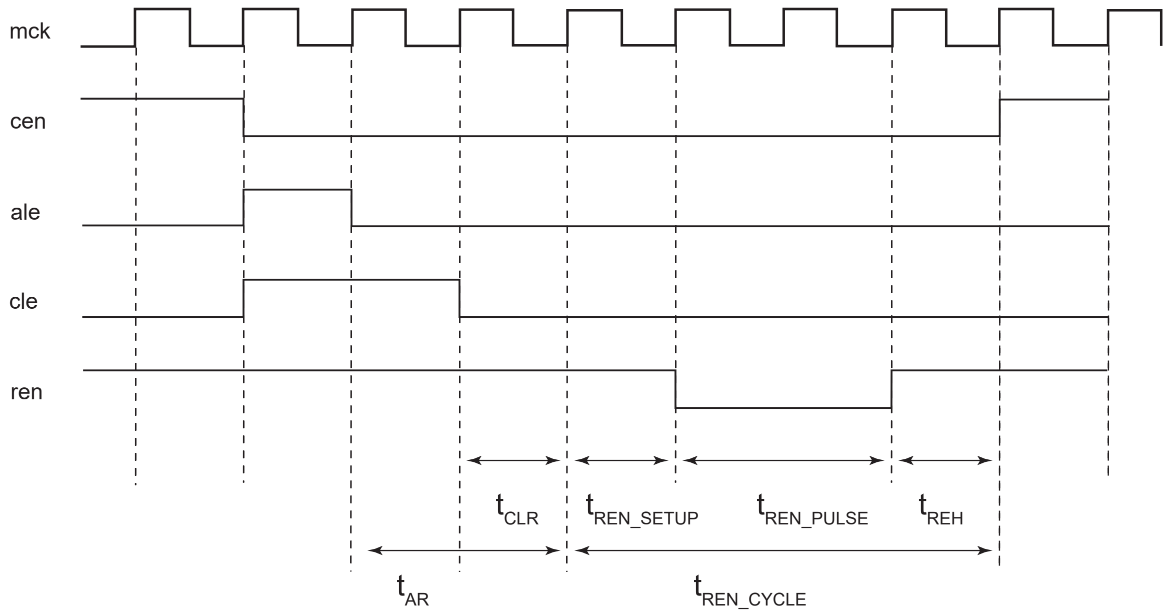
- Read Enable Configuration

Use NRD_SETUP, NRD_PULSE and NRD_CYCLE to define the read enable waveform according to the external device datasheet.

Use HSMC_TIMINGS.TAR to configure the timings between the address latch enable falling edge to read the enable falling edge.

Use HSMC_TIMINGS.TCLR to configure the timings between the command latch enable falling edge to read the enable falling edge.

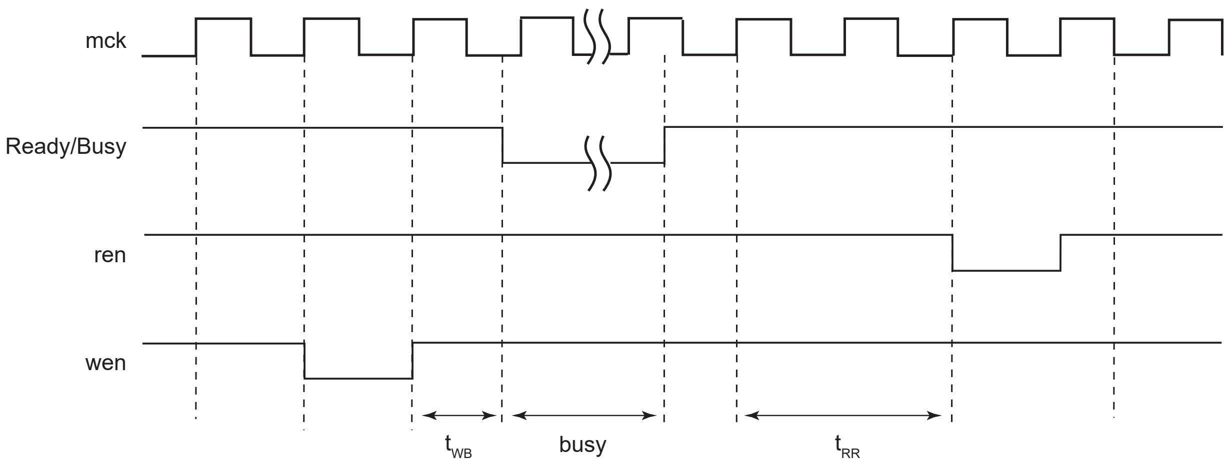
Figure 16-29. Read Enable Timing Configuration Working with NAND Flash Device



- Ready/Busy Signal Timing Configuration with a NAND Flash Device

Use HSMC_TIMINGS.TWB to configure the maximum elapsed time between the rising edge of the wen signal and the falling edge of the Ready/Busy signal. Use TRR field in the HSMC_TIMINGS register to program the number of clock cycles between the rising edge of the Ready/Busy signal and the falling edge of the ren signal.

Figure 16-30. Ready/Busy Timing Configuration

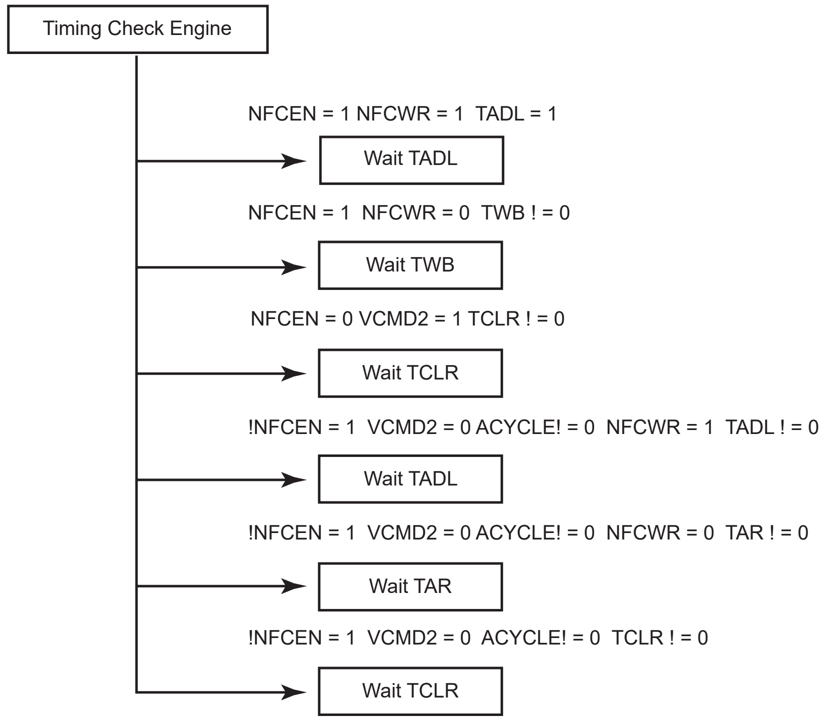


16.16.3.1 NFC Timing Engine

When the NFC Command register is written, the NFC issues a NAND Flash Command and optionally performs a data transfer between the NFC SRAM and the NAND Flash device. The NFC Timing Engine ensures valid NAND Flash timings, depending on the set of parameters decoded from the address bus. These timings are defined in the HSMC_TIMINGS register.

For information on the timing used depending on the command, see the figure below.

Figure 16-31. NFC Timing Engine



See the register descriptions NFCADDR_CMD and HSMC_TIMINGSx.

16.16.4 NFC SRAM

16.16.4.1 NFC SRAM Mapping

If the NFC is used to read and write data from and to the NAND Flash, the configuration depends on the page size (HSMC_CFG.PAGESIZE field). See the tables below for detailed mapping.

The NFC can handle the NAND Flash with a page size of 8 Kbytes or lower (2 Kbytes, for example). In case of a 4 Kbyte or lower page size, the NFC SRAM can be split into two banks. The HSMC_BANK.BANK bit is used to select where NAND Flash data are written or read. For an 8 Kbyte page size, this field is not relevant.

Note that a “Ping-Pong” mode (write or read to a bank while the NFC writes or reads to another bank) is accessible with the NFC (using two different banks).

If the NFC is not used, the NFC SRAM can be used for a general purpose by the application.

Table 16-8. NFC SRAM Bank Mapping for 512 bytes

Offset	Use	Access
0x00000000–0x000001FF	Main Area Bank 0	Read/Write
0x00000200–0x000003FF	Spare Area Bank 0	Read/Write
0x00001200–0x000013FF	Main Area Bank 1	Read/Write
0x00001400–0x000015FF	Spare Area Bank 1	Read/Write

Table 16-9. NFC SRAM Bank Mapping for 1 Kbyte

Offset	Use	Access
0x00000000–0x000003FF	Main Area Bank 0	Read/Write
0x00000400–0x000005FF	Spare Area Bank 0	Read/Write

...........continued

Offset	Use	Access
0x00001200–0x000015FF	Main Area Bank 1	Read/Write
0x00001600–0x000017FF	Spare Area Bank 1	Read/Write

Table 16-10. NFC SRAM Bank Mapping for 2 Kbytes

Offset	Use	Access
0x00000000–0x000007FF	Main Area Bank 0	Read/Write
0x00000800–0x000009FF	Spare Area Bank 0	Read/Write
0x00001200–0x000019FF	Main Area Bank 1	Read/Write
0x00001A00–0x00001BFF	Spare Area Bank 1	Read/Write

Table 16-11. NFC SRAM Bank Mapping for 4 Kbytes

Offset	Use	Access
0x00000000–0x00000FFF	Main Area Bank 0	Read/Write
0x00001000–0x000011FF	Spare Area Bank 0	Read/Write
0x00001200–0x000021FF	Main Area Bank 1	Read/Write
0x00002200–0x000023FF	Spare Area Bank 1	Read/Write

Table 16-12. NFC SRAM Bank Mapping for 8 Kbytes, only one bank is available

Offset	Use	Access
0x00000000–0x00001FFF	Main Area Bank 0	Read/Write
0x00002000–0x000023FF	Spare Area Bank 0	Read/Write

16.16.4.2 NFC SRAM Access Prioritization Algorithm

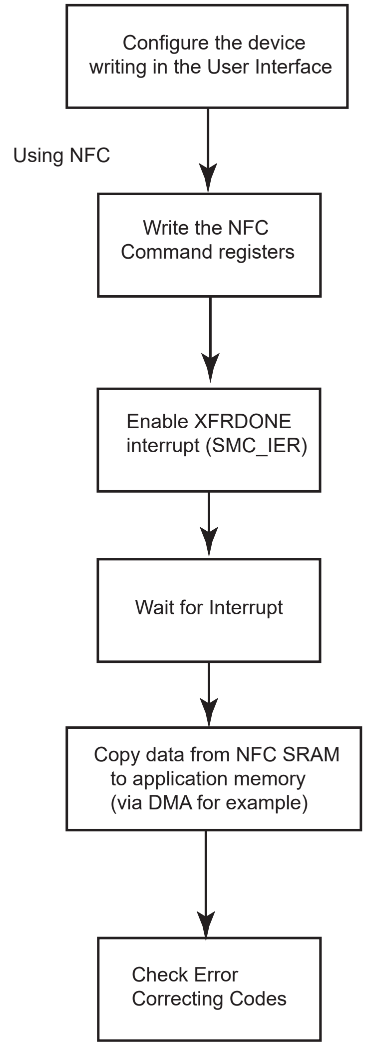
When the NFC is reading from or writing to an NFC SRAM bank, the other bank is available. If an NFC SRAM access occurs when the NFC performs a read or write operation in the same bank, then the access is discarded. The write operation is not performed. The read operation returns undefined data. If this situation is encountered, the AWB status flag located in the NFC Status Register is raised and indicates that a shared resource access violation has occurred.

16.16.5 NAND Flash Operations

This section describes the software operations needed to issue commands to the NAND Flash device and to perform data transfers using the NFC.

16.16.5.1 Page Read

Figure 16-32. Page Read Flow Chart

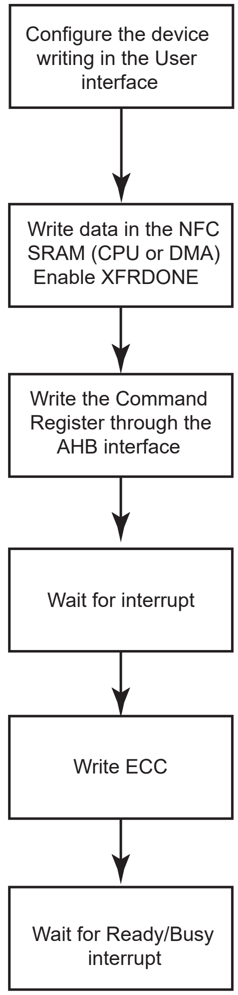


Instead of using the interrupt, the NFCBUSY flag can be polled.

For more information on the NFC Control register, see NFCADDR_CMD.

16.16.5.2 Program Page

Figure 16-33. Program Page Flow Chart



Writing the ECC cannot be done using the NFC; it needs to be done “manually”.

Instead of using the interrupt, the NFCBUSY flag can be polled.

For more information on the NFC Control register, see NFCADDR_CMD.

16.17 PMECC Controller Functional Description

The Programmable Multibit Error Correcting Code (PMECC) controller is a programmable binary BCH (Bose, Chaudhuri and Hocquenghem) encoder/decoder. This controller can be used to generate redundancy information for both SLC and MLC NAND devices. It supports redundancy for correction of 2, 4, 8, 12, 24 or 32 errors per sector of data. The sector size is programmable and can be set to 512 bytes or 1024 bytes. The PMECC module generates redundancy at encoding time, when a NAND write page operation is performed. The redundancy is appended to the page and written in the spare area. This operation is performed by the processor. It moves the content of the PMECCX registers into the NAND Flash memory. The number of registers depends on the selected error correction capability (see the table Relevant Redundancy Registers). This operation is executed for each sector. At decoding time, the PMECC module generates the remainders of the received codeword by the minimal polynomials. When all remainders for a given sector are set to zero, no error occurred. When the remainders are different from zero, the codeword is corrupted and further processing is required.

The PMECC module generates an interrupt indicating that an error occurred. The processor must read the PMECC Interrupt Status register (HSMC_PMECCISR). This register indicates which sector is corrupted.

The processor must execute the following decoding steps to find the error location within a sector:

1. Compute syndrome
2. Find the error location polynomial
3. Find the roots of the error location polynomial

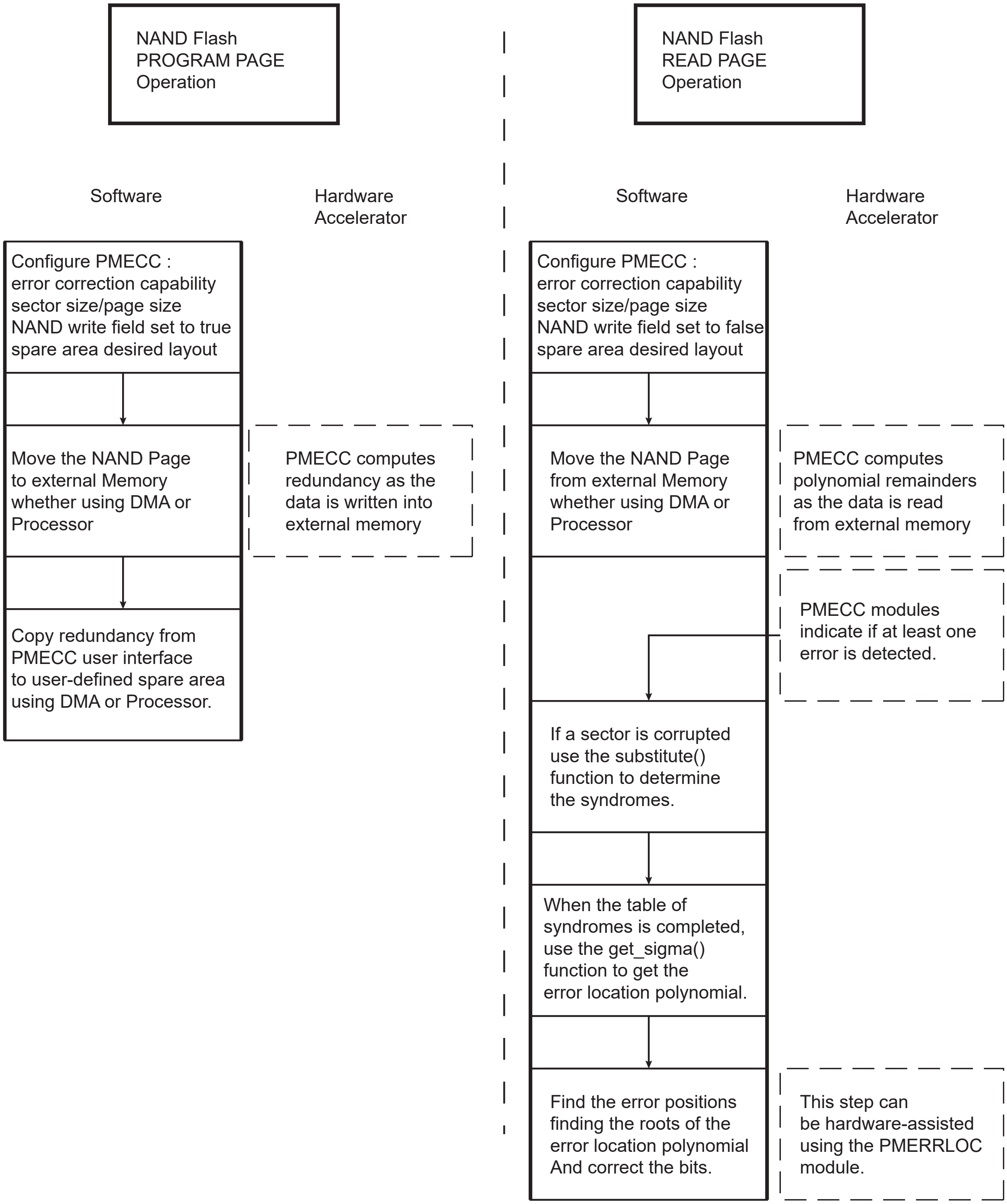
All decoding steps involve finite field computation. It means that a library of finite field arithmetic must be available to perform addition, multiplication and inversion. These arithmetic operations can be performed through the use of a memory mapped lookup table, or direct software implementation. The software implementation presented is based on lookup tables. Two tables named gf_log and gf_antilog are used. If alpha is the primitive element of the field, then a power of alpha is in the field. Assuming that beta = alpha ^ index, then beta belongs to the field, and gf_log(beta) = gf_log(alpha ^ index) = index. The gf_antilog table provides exponent inverse of the element; if beta = alpha ^ index, then gf_antilog(index) = beta.

The first step consists in the syndrome computation. The PMECC module computes the remainders and the software must substitute the power of the primitive element. The procedure implementation is given in the section Remainder Substitution Procedure.

The second step is the most software intensive. It is the Berlekamp's iterative algorithm for finding the error-location polynomial. The procedure implementation is given in the section Finding the Error Location Polynomial Sigma(x).

The Last step is finding the root of the error location polynomial. This step can be very software intensive. Indeed there is no straightforward method of finding the roots, except evaluating each element of the field in the error location polynomial. However, a hardware accelerator can be used to find the roots of the polynomial. The PMERRLOC module provides this kind of hardware acceleration.

Figure 16-34. Software Hardware Multibit Error Correction Dataflow



16.17.1 MLC/SLC Write Page Operation Using PMECC

When an MLC write page operation is performed, the PMECC controller is configured with the NANDWR bit of the PMECC Configuration (HSMC_PMECCFG) register set to one. When the NAND spare area contains file system information and redundancy (PMECCx), the spare area is error-protected, then the HSMC_PMECCFG.SPAREEN bit is set. When the NAND spare area contains only redundancy information, the SPAREEN bit is cleared.

When the write page operation is terminated, the user writes the redundancy in the NAND spare area. This operation can be done with DMA assistance.

Table 16-13. Relevant Redundancy Registers

BCH_ERR Field	Sector Size Set to 512 Bytes	Sector Size Set to 1024 Bytes
0	PMECC0	PMECC0
1	PMECC0, PMECC1	PMECC0, PMECC1
2	PMECC0, PMECC1, PMECC2, PMECC3	PMECC0, PMECC1, PMECC2, PMECC3
3	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6
4	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6, PMECC7, PMECC8, PMECC9	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6, PMECC7, PMECC8, PMECC9, PMECC10
5	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6, PMECC7, PMECC8, PMECC9, PMECC10, PMECC11, PMECC12	PMECC0, PMECC1, PMECC2, PMECC3, PMECC4, PMECC5, PMECC6, PMECC7, PMECC8, PMECC9, PMECC10, PMECC11, PMECC12, PMECC13

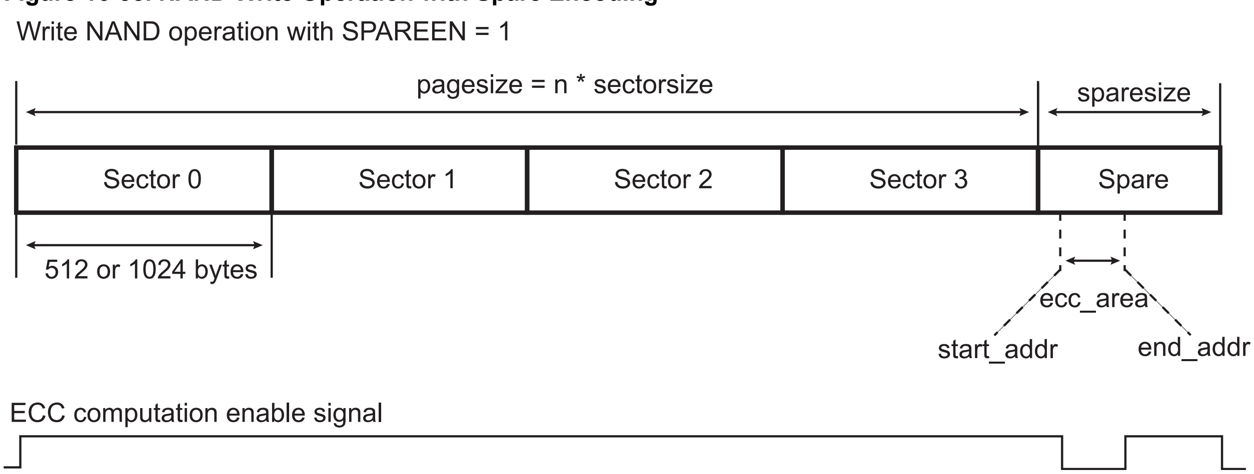
Table 16-14. Number of Relevant ECC Bytes per Sector, Copied from LSByte to MSByte

BCH_ERR Field	Sector Size Set to 512 Bytes	Sector Size Set to 1024 Bytes
0	4 bytes	4 bytes
1	7 bytes	7 bytes
2	13 bytes	14 bytes
3	20 bytes	21 bytes
4	39 bytes	42 bytes
5	52 bytes	56 bytes

16.17.1.1 SLC/MLC Write Operation with Spare Enable Bit Set

When PMECCFG.SPAREEN is set, the spare area of the page is encoded with the data stream of the last sector of the page. This mode is entered by setting PMECCTRL.DATA. When the encoding process is over, the redundancy is written to the spare area in User mode. PMECCTRL.USER must be set.

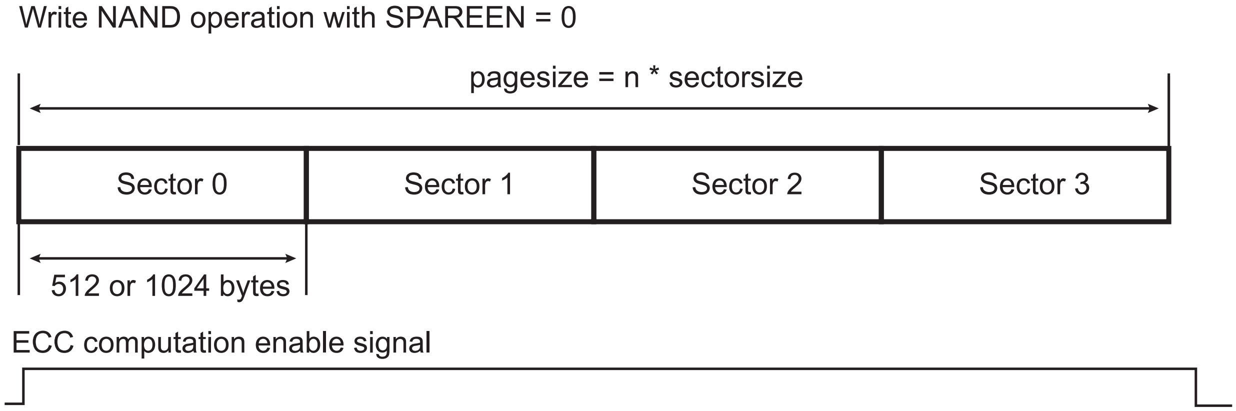
Figure 16-35. NAND Write Operation with Spare Encoding



16.17.1.2 SLC/MLC Write Operation with Spare Disable

When PMECCFG.SPAREEN is cleared, the spare area is not encoded with the data stream. This mode is entered by setting PMECCTRL.DATA.

Figure 16-36. NAND Write Operation



16.17.2 MLC/SLC Read Page Operation Using PMECC

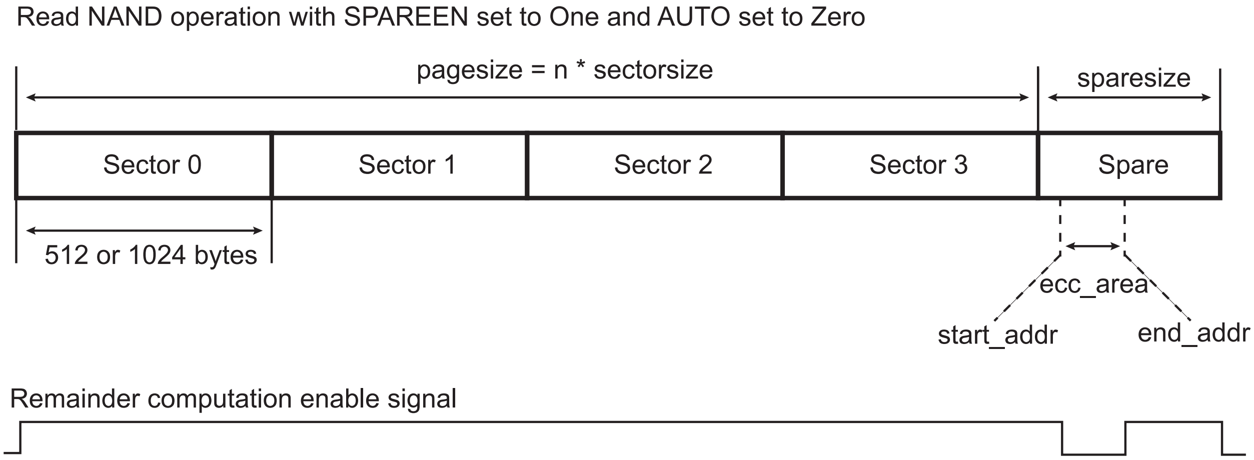
Table 16-15. Relevant Remainder Registers

BCH_ERR Field	Sector Size Set to 512 Bytes	Sector Size Set to 1024 Bytes
0	PMECCREM0	PMECCREM0
1	PMECCREM0, PMECCREM1	PMECCREM0, PMECCREM1
2	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3
3	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7
4	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7, PMECCREM8, PMECCREM9, PMECCREM10, PMECCREM11	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7, PMECCREM8, PMECCREM9, PMECCREM10, PMECCREM11
5	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7, PMECCREM8, PMECCREM9, PMECCREM10, PMECCREM11, PMECCREM12, PMECCREM13, PMECCREM14, PMECCREM15	PMECCREM0, PMECCREM1, PMECCREM2, PMECCREM3, PMECCREM4, PMECCREM5, PMECCREM6, PMECCREM7, PMECCREM8, PMECCREM9, PMECCREM10, PMECCREM11, PMECCREM12, PMECCREM13, PMECCREM14, PMECCREM15

16.17.2.1 MLC/SLC Read Operation with Spare Decoding

When the spare area is protected, it contains valid data. As redundancy may be included in the middle of the information stream, program the start address and the end address of the ECC area. The controller will automatically skip the ECC area. To enter this mode, write a 1 in HSMC_PMECCTRL.DATA. When the page is fully retrieved from the NAND, the ECC area is read in User mode (write a 1 to PMECCTRL.USER to enter User mode).

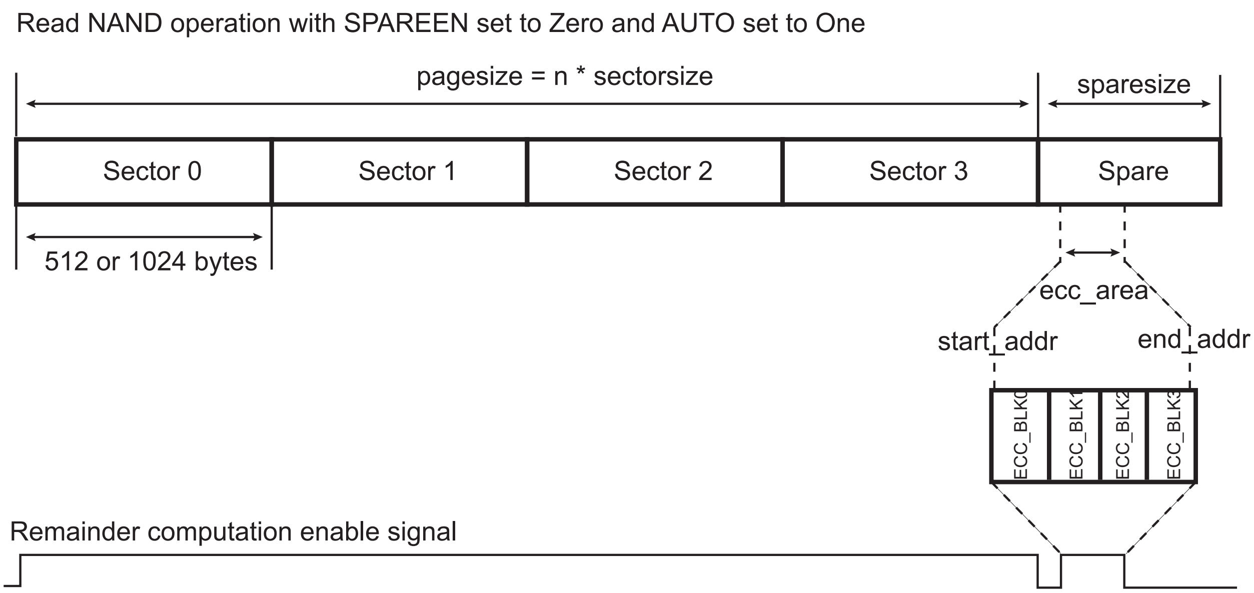
Figure 16-37. Read Operation with Spare Decoding



16.17.2.2 MLC/SLC Read Operation

If the spare area is not protected with the error correcting code, the redundancy area is retrieved directly. To enter this mode, write a 1 in PMECCTRL.DATA. When the AUTO field is set to one, the ECC is retrieved automatically; otherwise, read ECC via User mode.

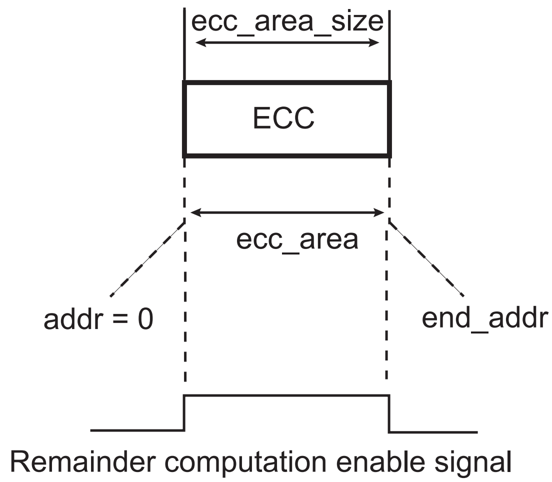
Figure 16-38. Read Operation



16.17.2.3 MLC/SLC User Read ECC Area

This mode allows a manual retrieve of the ECC. To enter the mode, write a 1 in PMECCTRL.USER.

Figure 16-39. Read User Mode



16.17.2.4 MLC Controller Working with NFC

Table 16-16. MLC Controller Configuration when the Host Controller is Used

Transfer Type	NFC		PMECC		
	RSPARE	WSPARE	SPAREEN	AUTO	User Mode
Program Page main area is protected, spare is not protected, spare is written manually	0	0	0	0	Not used
Program Page main area is protected, spare is protected, spare is written by NFC	0	1	1	0	Not used
Read Page main area is protected, spare is not protected, spare is not retrieved by NFC	0	0	0	0	Used
Read Page main area is protected, spare is not protected, spare is retrieved by NFC	1	0	0	1	Not used
Read Page main area is protected, spare is protected, spare is retrieved by NFC	1	0	1	0	Used

16.18 Software Implementation

16.18.1 Remainder Substitution Procedure

The substitute function evaluates the remainder polynomial, with different values of the field primitive element. The addition arithmetic operation is performed with the exclusive OR. The multiplication arithmetic operation is performed through the gf_log and gf_antilog lookup tables.

The REM2NP1 and REMN2NP3 fields of the PMECCREMN registers contain only odd remainders. Each bit indicates whether the coefficient of the remainder polynomial is set to zero or not.

NB_ERROR_MAX defines the maximum value of the error correcting capability.

NB_ERROR defines the error correcting capability selected at encoding/decoding time.

NB_FIELD_ELEMENTS defines the number of elements in the field.

si[] is a table that holds the current syndrome value. An element of that table belongs to the field. This is also a shared variable for the next step of the decoding operation.

oo[] is a table that contains the degree of the remainders.

```
int substitute()
{
int i;
int j;
for (i = 1; i < 2 * NB_ERROR_MAX; i++)
{
   si[i] = 0;
}
for (i = 1; i < 2*NB_ERROR; i++)
{
   for (j = 0; j < oo[i]; j++)
 {
     if (REM2NPX[i][j])
    {
      si[i] = gf_antilog[(i * j)%NB_FIELD_ELEMENTS] ^ si[i];
     }
   }
}
return 0;
}
```

16.18.2 Finding the Error Location Polynomial Sigma(x)

The sample code below gives a Berlekamp iterative procedure for finding the value of the error location polynomial.

The input of the procedure is the si[] table defined in the remainder substitution procedure.

The output of the procedure is the error location polynomial named smu (sigma mu). The polynomial coefficients belong to the field. The smu[NB_ERROR+1][] is a table that contains all these coefficients.

NB_ERROR_MAX defines the maximum value of the error correcting capability.

NB_ERROR defines the error correcting capability selected at encoding/decoding time.

NB_FIELD_ELEMENTS defines the number of elements in the field.

```
int get_sigma()
    {
    int i;
    int j;
    int k;
    /* mu          */
    int mu[NB_ERROR_MAX+2];
    /* sigma ro    */
    int sro[2*NB_ERROR_MAX+1];
    /* discrepancy */
    int dmu[NB_ERROR_MAX+2];
    /* delta order   */
    int delta[NB_ERROR_MAX+2];
    /* index of largest delta */
    int ro;
    int largest;
    int diff;
    /*                   */
    /*     First Row     */
    /*                   */
    /* Mu */
    mu[0]  = -1; /* Actually -1/2  */
    /* Sigma(x) set to 1  */
    for (i = 0; i < (2*NB_ERROR_MAX+1); i++)
     smu[0][i] = 0;
    smu[0][0] = 1;
    /* discrepancy set to 1 */
    dmu[0] = 1;
    /* polynom order set to 0 */
    lmu[0] = 0;
    /* delta set to -1 */
    delta[0]  = (mu[0] * 2 - lmu[0]) >> 1;
    /*                  */
```

```
    /*     Second Row   */
    /*                  */
    /* Mu */
    mu[1]  = 0;
    /* Sigma(x) set to 1 */
    for (i = 0; i < (2*NB_ERROR_MAX+1); i++)
     smu[1][i] = 0;
    smu[1][0] = 1;
    /* discrepancy set to Syndrome 1 */
    dmu[1] = si[1];
    /* polynom order set to 0 */
    lmu[1] = 0;
    /* delta set to 0 */
    delta[1]  = (mu[1] * 2 - lmu[1])>> 1;
    for (i=1; i <= NB_ERROR; i++)
    {
     mu[i+1] = i << 1;
    /*************************************************/
     /*                                              */
     /*                                              */
     /*          Compute Sigma (Mu+1)                */
     /*          And L(mu)                           */
     /* check if discrepancy is set to 0 */
     if (dmu[i] == 0)
     {
     /* copy polynom */
     for (j=0; j<2*NB_ERROR_MAX+1; j++)
     {
      smu[i+1][j] =
      smu[i][j];
     }
     /* copy previous polynom order to the next */
     lmu[i+1] = lmu[i];
     }
     else
     {
     ro      = 0;
     largest = -1;
     /* find largest delta with dmu != 0 */
     for (j=0; j<i; j++)
      {
      if (dmu[j])
       {
       if (delta[j] > largest)
        {
        largest = delta[j];
        ro   = j;
        }
       }
      }
     /* initialize signal ro */
     for (k = 0; k < 2*NB_ERROR_MAX+1; k ++)
     {
      sro[k] = 0;
     }
     /* compute difference */
     diff = (mu[i] - mu[ro]);
     /* compute X ^ (2(mu-ro)) */
     for (k = 0; k < (2*NB_ERROR_MAX+1); k ++)
     {
      sro[k+diff] =  smu[ro][k];
     }
     /* multiply by dmu * dmu[ro]^-1 */
     for (k = 0; k < 2*NB_ERROR_MAX+1; k ++)
     {
      /* dmu[ro] is not equal to zero by definition */
      /* check that operand are different from 0    */
      if (sro[k] && dmu[i])
      {
       /* galois inverse  */
       sro[k] = gf_antilog[(gf_log[dmu[i]] + (NB_FIELD_ELEMENTS-gf_log[dmu[ro]]) +
gf_log[sro[k]]) %
  NB_FIELD_ELEMENTS];
      }
     }
     /* multiply by dmu * dmu[ro]^-1 */
```

```
      for (k = 0; k < 2*NB_ERROR_MAX+1; k++)
      {
       smu[i+1][k] = smu[i][k] ^ sro[k];
       if (smu[i+1][k])
       {
        /* find the order of the polynom */
        lmu[i+1] = k << 1;
       }
      }
     }
     /*                                                */
     /*                                                */
     /*      End Compute Sigma (Mu+1)                  */
     /*      And L(mu)                                 */
     /*************************************************/
     /* In either case compute delta  */
     delta[i+1]  = (mu[i+1] * 2 - lmu[i+1]) >> 1;
     /* In either case compute the discrepancy */
     for (k = 0 ; k <= (lmu[i+1]>>1); k++)
     {
      if (k == 0)
      dmu[i+1] = si[2*(i-1)+3];
      /* check if one operand of the multiplier is null, its index is -1 */
      else if (smu[i+1][k] && si[2*(i-1)+3-k])
      dmu[i+1] = gf_antilog[(gf_log[smu[i+1][k]] + gf_log[si[2*(i-1)+3-k]])%nn] ^ dmu[i+1];
     }
     }
     return 0;
     }
```

16.18.3 Finding the Error Position

The output of the get_sigma() procedure is a polynomial stored in the smu[NB_ERROR+1][] table. The error positions are the roots of that polynomial. The degree of that polynomial is a very important information, as it gives the number of errors. The PMERRLOC module provides hardware accelerator for that step.

16.18.3.1 Error Location

The PMECC Error Location controller provides hardware acceleration for determining roots of polynomials over two finite fields: GF(2^13) and GF(2^14). It integrates 32 fully programmable coefficients. These coefficients belong to GF(2^13) or GF(2^14). The coefficient programmed in PMERRLOC{i} is the coefficient of X ^ i in the polynomial.

The search operation is started as soon as a write access is detected in the PMECC Error Location Enable (HSMC_ELEN) register and can be disabled by writing to the PMECC Error Location Disable (HSMC_ELDIS) register. Initialize HSMC_ELEN.ENINIT with the number of galois field elements to test.

Table 16-17. ENINIT Field Value for a Sector Size of 512 Bytes

Error Correcting Capability	ENINIT Value
2	4122
4	4148
8	4200
12	4252
24	4408
32	4512

Table 16-18. ENINIT Field Value for a Sector Size of 1024 Bytes

Error Correcting Capability	ENINIT Value
2	8220
4	8248
8	8304

...........continued

Error Correcting Capability	ENINIT Value
12	8360
24	8528
32	8640

While the PMECC engine searches for roots, the BUSY field in the PMECC Error Location Status register (HSMC_ELSR) remains asserted. An interrupt is asserted at the end of the computation, and the DONE bit in the PMECC Error Location Interrupt Status register (HSMC_ELSIR) is set. The HSMC_ELISR.ERR_CNT field shows the number of errors. The PMECC Error Location x registers (HSMC_ERRLOCx) show the error position.

16.19 Register Summary

Notes: The following blocks of registers are instanced 8 times in the user interface:

- HSMC_PMECCx[x=0..13]
- HSMCC_REMx[x=0..15]

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	HSMC_CFG	31:24		NFCSPARESIZE[6:0]						
		23:16		DTOMUL[2:0]			DTOCYC[3:0]			
		15:8			RBEDGE	EDGECTRL			RSPARE	WSPARE
		7:0						PAGESIZE[2:0]		
0x04	HSMC_CTRL	31:24								
		23:16								
		15:8								
		7:0							NFCDIS	NFCEN
0x08	HSMC_SR	31:24								RB_EDGE0
		23:16	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
		15:8		NFCSID[2:0]			NFCWR			NFCBUSY
		7:0			RB_FALL	RB_RISE				SMCSTS
0x0C	HSMC_IER	31:24								RB_EDGE0
		23:16	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
		15:8								
		7:0			RB_FALL	RB_RISE				
0x10	HSMC_IDR	31:24								RB_EDGE0
		23:16	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
		15:8								
		7:0			RB_FALL	RB_RISE				
0x14	HSMC_IMR	31:24								RB_EDGE0
		23:16	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
		15:8								
		7:0			RB_FALL	RB_RISE				
0x18	HSMC_ADDR	31:24								
		23:16								
		15:8								
		7:0	ADDR_CYCLE0[7:0]							
0x1C	HSMC_BANK	31:24								
		23:16								
		15:8								
		7:0								BANK
0x20 ... 0x6F	Reserved									
0x70	HSMC_PMECCFG	31:24								
		23:16				AUTO				SPAREEN
		15:8				NANDWR			PAGESIZE[1:0]	
		7:0				SECTORSZ		BCH_ERR[2:0]		
0x74	HSMC_PMECCSAREA	31:24								
		23:16								
		15:8								SPARESIZE[8]
		7:0	SPARESIZE[7:0]							
0x78	HSMC_PMECCSADDR	31:24								
		23:16								
		15:8								STARTADDR[8]
		7:0	STARTADDR[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x7C	HSMC_PMECCEADDR	31:24								
		23:16								
		15:8								ENDADDR[8]
		7:0	ENDADDR[7:0]							
0x80 ... 0x83	Reserved									
0x84	HSMC_PMECCTRL	31:24								
		23:16								
		15:8								
		7:0			DISABLE	ENABLE		USER	DATA	RST
0x88	HSMC_PMECCSR	31:24								
		23:16								
		15:8								
		7:0				ENABLE				BUSY
0x8C	HSMC_PMECCIER	31:24								
		23:16								
		15:8								
		7:0								ERRIE
0x90	HSMC_PMECCIDR	31:24								
		23:16								
		15:8								
		7:0								ERRID
0x94	HSMC_PMECCIMR	31:24								
		23:16								
		15:8								
		7:0								ERRIM
0x98	HSMC_PMECCISR	31:24								
		23:16								
		15:8								
		7:0	ERRIS[7:0]							
0x9C ... 0xAF	Reserved									
0xB0	HSMC_PMECC0	31:24	ECC[31:24]							
		23:16	ECC[23:16]							
		15:8	ECC[15:8]							
		7:0	ECC[7:0]							
...										
0xE4	HSMC_PMECC13	31:24	ECC[31:24]							
		23:16	ECC[23:16]							
		15:8	ECC[15:8]							
		7:0	ECC[7:0]							
0xE8 ... 0x02AF	Reserved									
0x02B0	HSMC_REM0	31:24			REM2NP3[13:8]					
		23:16	REM2NP3[7:0]							
		15:8			REM2NP1[13:8]					
		7:0	REM2NP1[7:0]							
...										
0x02EC	HSMC_REM15	31:24			REM2NP3[13:8]					
		23:16	REM2NP3[7:0]							
		15:8			REM2NP1[13:8]					
		7:0	REM2NP1[7:0]							
0x02F0 ... 0x04FF	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0500	HSMC_ELCFG	31:24								
		23:16				ERRNUM[4:0]				
		15:8								
		7:0								SECTORSZ
0x0504	HSMC_ELPRIM	31:24								
		23:16								
		15:8	PRIMITIV[15:8]							
		7:0	PRIMITIV[7:0]							
0x0508	HSMC_ELEN	31:24								
		23:16								
		15:8			ENINIT[13:8]					
		7:0	ENINIT[7:0]							
0x050C	HSMC_ELDIS	31:24								
		23:16								
		15:8								
		7:0								DIS
0x0510	HSMC_ELSR	31:24								
		23:16								
		15:8								
		7:0								BUSY
0x0514	HSMC_ELIER	31:24								
		23:16								
		15:8								
		7:0								DONE
0x0518	HSMC_ELIDR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x051C	HSMC_ELIMR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x0520	HSMC_ELISR	31:24								
		23:16								
		15:8			ERR_CNT[5:0]					
		7:0								DONE
0x0524 ... 0x0527	Reserved									
0x0528	HSMC_SIGMA0	31:24								
		23:16								
		15:8			SIGMA0[13:8]					
		7:0	SIGMA0[7:0]							
0x052C	HSMC_SIGMA1	31:24								
		23:16								
		15:8			SIGMAx[13:8]					
		7:0	SIGMAx[7:0]							
...										
0x05A8	HSMC_SIGMA32	31:24								
		23:16								
		15:8			SIGMAx[13:8]					
		7:0	SIGMAx[7:0]							
0x05AC	HSMC_ERRLOC0	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							
...										

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0628	HSMC_ERRLOC31	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							
0x062C ... 0x06FF	Reserved									
0x0700	HSMC_SETUP0	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x0704	HSMC_PULSE0	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x0708	HSMC_CYCLE0	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x070C	HSMC_TIMINGS0	31:24	NFSEL				TWB[3:0]			
		23:16					TRR[3:0]			
		15:8				OCMS	TAR[3:0]			
		7:0	TADL[3:0]				TCLR[3:0]			
0x0710	HSMC_MODE0	31:24								
		23:16								
		15:8				DBW				BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x0714	HSMC_SETUP1	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x0718	HSMC_PULSE1	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x071C	HSMC_CYCLE1	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x0720	HSMC_TIMINGS1	31:24	NFSEL				TWB[3:0]			
		23:16					TRR[3:0]			
		15:8				OCMS	TAR[3:0]			
		7:0	TADL[3:0]				TCLR[3:0]			
0x0724	HSMC_MODE1	31:24								
		23:16								
		15:8				DBW				BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x0728	HSMC_SETUP2	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x072C	HSMC_PULSE2	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x0730	HSMC_CYCLE2	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x0734	HSMC_TIMINGS2	31:24	NFSEL				TWB[3:0]			
		23:16					TRR[3:0]			
		15:8				OCMS	TAR[3:0]			
		7:0	TADL[3:0]				TCLR[3:0]			
0x0738	HSMC_MODE2	31:24								
		23:16								
		15:8				DBW				BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x073C	HSMC_SETUP3	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x0740	HSMC_PULSE3	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x0744	HSMC_CYCLE3	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x0748	HSMC_TIMINGS3	31:24	NFSEL				TWB[3:0]			
		23:16					TRR[3:0]			
		15:8				OCMS	TAR[3:0]			
		7:0	TADL[3:0]				TCLR[3:0]			
0x074C	HSMC_MODE3	31:24								
		23:16								
		15:8				DBW				BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x0750 ... 0x079F	Reserved									
0x07A0	HSMC_OCMS	31:24								
		23:16								
		15:8								
		7:0							SRSE	SMSE
0x07A4	HSMC_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x07A8	HSMC_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x07AC	HSMC_CLKCFG	31:24								
		23:16							CLKEDGE	CLKEN
		15:8								CLKDIV[8]
		7:0	CLKDIV[7:0]							
0x07B0 ... 0x07E3	Reserved									
0x07E4	HSMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0x07E8	HSMC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

16.19.1 NFC Configuration Register

Name: HSMC_CFG
Offset: 0x000
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		NFCSPARESIZE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DTOMUL[2:0]			DTOCYC[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RBEDGE	EDGECTRL			RSPARE	WSPARE
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
						PAGESIZE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 30:24 – NFCSPARESIZE[6:0] NAND Flash Spare Area Size Retrieved by the Host Controller
The spare size is set to (NFCSPARESIZE + 1) * 4 bytes. The spare area is only retrieved when RSPARE or WSPARE is activated.

Bits 22:20 – DTOMUL[2:0] Data Timeout Multiplier
These fields determine the maximum number of main system bus clock cycles that the SMC waits until the detection of a rising edge on Ready/Busy signal.
If the data timeout set by DTOCYC and DTOMUL has been exceeded, the Data Timeout Error flag (DTOE) in the NFC Status Register (NFC_SR) raises.
Data Timeout Multiplier is defined by DTOMUL as shown in the following table:

Value	Name	Description
0	X1	DTOCYC
1	X16	DTOCYC x 16
2	X128	DTOCYC x 128
3	X256	DTOCYC x 256
4	X1024	DTOCYC x 1024
5	X4096	DTOCYC x 4096
6	X65536	DTOCYC x 65536
7	X1048576	DTOCYC x 1048576

Bits 19:16 – DTOCYC[3:0] Data Timeout Cycle Number

Bit 13 – RBEDGE Ready/Busy Signal Edge Detection

Value	Description
0	Indicates the level of the Ready/Busy line.
1	Indicates that a transition has occurred on the Ready/Busy line.

Bit 12 – EDGECTRL Rising/Falling Edge Detection Control

Value	Description
0	Rising edge is detected.
1	Falling edge is detected.

Bit 9 – RSPARE Read Spare Area

Value	Description
0	The NFC skips the spare area in Read mode.
1	The NFC reads both main area and spare area in Read mode.

Bit 8 – WSPARE Write Spare Area

Value	Description
0	The NFC skips the spare area in Write mode.
1	The NFC writes both main area and spare area in Write mode.

Bits 2:0 – PAGESIZE[2:0] Page Size of the NAND Flash Device

Value	Name	Description
0	PS512	Main area 512 bytes
1	PS1024	Main area 1024 bytes
2	PS2048	Main area 2048 bytes
3	PS4096	Main area 4096 bytes
4	PS8192	Main area 8192 bytes

16.19.2 NFC Control Register

Name: HSMC_CTRL
Offset: 0x004
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							NFCDIS	NFCEN
Access							W	W
Reset							–	–

Bit 1 – NFCDIS NAND Flash Controller Disable

Value	Description
0	No effect.
1	Disables the NAND Flash controller.

Bit 0 – NFCEN NAND Flash Controller Enable

Value	Description
0	No effect.
1	Enables the NAND Flash controller.

16.19.3 NFC Status Register

Name: HSMC_SR
Offset: 0x008
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								RB_EDGE0
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
Access	R	R	R	R			R	R
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
		NFCSID[2:0]			NFCWR			NFCBUSY
Access		R	R	R	R			R
Reset		0	0	0	0			0

Bit	7	6	5	4	3	2	1	0
			RB_FALL	RB_RISE				SMCSTS
Access			R	R				R
Reset			0	0				0

Bit 24 – RB_EDGE0 Ready/Busy Line 0 Edge Detected
If set to one, this flag indicates that an edge has been detected on the Ready/Busy Line 0. Depending on HSMC_CFG.EDGECTRL, only the rising or the falling edge is detected. This flag is reset after a Status Read operation.

Bit 23 – NFCASE NFC Access Size Error
If set to one, this flag indicates that an illegal access has been detected in the NFC Memory Area. Only Word access is allowed within the NFC memory area. This flag is reset after a Status Read operation.

Bit 22 – AWB Accessing While Busy
If set to one, this flag indicates that an AHB host has performed an access during the busy phase. This flag is reset after a Status Read operation.

Bit 21 – UNDEF Undefined Area Error
When set to one, this flag indicates that the processor performed an access in an undefined memory area. This flag is reset after a Status Read operation.

Bit 20 – DTOE Data Timeout Error
When set to one, this flag indicates that the data timeout set by DTOMUL and DTOCYC has been exceeded. This flag is reset after a Status Read operation.

Bit 17 – CMDDONE Command Done
When set to one, this flag indicates that the NFC has terminated the command. This flag is reset after a Status Read operation.

Bit 16 – XFRDONE NFC Data Transfer Terminated
When set to one, this flag indicates that the NFC has terminated the data transfer. This flag is reset after a Status Read operation.

Bits 14:12 – NFCSID[2:0] NFC Chip Select ID (this field cannot be reset)
When a command is issued, this field indicates the value of the targeted chip select.

Bit 11 – NFCWR NFC Write/Read Operation (this field cannot be reset)
When a command is issued, this field indicates the current Read or Write operation.

Bit 8 – NFCBUSY NFC Busy (this field cannot be reset)
When set to one, this flag indicates that the controller is activated and accesses the memory device.

Bit 5 – RB_FALL Selected Ready Busy Falling Edge Detected
When set to one, this flag indicates that a falling edge on the Ready/Busy Line has been detected. This flag is reset after a Status Read operation. The Ready/Busy line is selected through the decoding of field HSMC_SR.NFCSID.

Bit 4 – RB_RISE Selected Ready Busy Rising Edge Detected
When set to one, this flag indicates that a rising edge on the Ready/Busy Line has been detected. This flag is reset after a Status Read operation. The Ready/Busy line is selected through the decoding of field HSMC_SR.NFCSID.

Bit 0 – SMCSTS NAND Flash Controller Status (this field cannot be reset)

Value	Description
0	The NAND Flash controller is disabled.
1	The NAND Flash controller is enabled.

16.19.4 NFC Interrupt Enable Register

Name: HSMC_IER
Offset: 0x00C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								RB_EDGE0
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
Access	W	W	W	W			W	W
Reset	–	–	–	–			–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RB_FALL	RB_RISE				
Access			W	W				
Reset			–	–				

Bit 24 – RB_EDGE0 Ready/Busy Line 0 Interrupt Enable

Bit 23 – NFCASE NFC Access Size Error Interrupt Enable

Bit 22 – AWB Accessing While Busy Interrupt Enable

Bit 21 – UNDEF Undefined Area Access Interrupt Enable

Bit 20 – DTOE Data Timeout Error Interrupt Enable

Bit 17 – CMDDONE Command Done Interrupt Enable

Bit 16 – XFRDONE Transfer Done Interrupt Enable

Bit 5 – RB_FALL Ready Busy Falling Edge Detection Interrupt Enable

Bit 4 – RB_RISE Ready Busy Rising Edge Detection Interrupt Enable

16.19.5 NFC Interrupt Disable Register

Name: HSMC_IDR
Offset: 0x010
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								RB_EDGE0
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
Access	W	W	W	W			W	W
Reset	–	–	–	–			–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RB_FALL	RB_RISE				
Access			W	W				
Reset			–	–				

Bit 24 – RB_EDGE0 Ready/Busy Line 0 Interrupt Disable

Bit 23 – NFCASE NFC Access Size Error Interrupt Disable

Bit 22 – AWB Accessing While Busy Interrupt Disable

Bit 21 – UNDEF Undefined Area Access Interrupt Disable

Bit 20 – DTOE Data Timeout Error Interrupt Disable

Bit 17 – CMDDONE Command Done Interrupt Disable

Bit 16 – XFRDONE Transfer Done Interrupt Disable

Bit 5 – RB_FALL Ready Busy Falling Edge Detection Interrupt Disable

Bit 4 – RB_RISE Ready Busy Rising Edge Detection Interrupt Disable

16.19.6 NFC Interrupt Mask Register

Name: HSMC_IMR
Offset: 0x014
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: Disables the corresponding interrupt.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								RB_EDGE0
Access								R
Reset								0
Bit	23	22	21	20	19	18	17	16
	NFCASE	AWB	UNDEF	DTOE			CMDDONE	XFRDONE
Access	R	R	R	R			R	R
Reset	0	0	0	0			0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RB_FALL	RB_RISE				
Access			R	R				
Reset			0	0				

Bit 24 – RB_EDGE0 Ready/Busy Line 0 Interrupt Mask

Bit 23 – NFCASE NFC Access Size Error Interrupt Mask

Bit 22 – AWB Accessing While Busy Interrupt Mask

Bit 21 – UNDEF Undefined Area Access Interrupt Mask5

Bit 20 – DTOE Data Timeout Error Interrupt Mask

Bit 17 – CMDDONE Command Done Interrupt Mask

Bit 16 – XFRDONE Transfer Done Interrupt Mask

Bit 5 – RB_FALL Ready Busy Falling Edge Detection Interrupt Mask

Bit 4 – RB_RISE Ready Busy Rising Edge Detection Interrupt Mask

16.19.7 NFC Address Cycle Zero Register

Name: HSMC_ADDR
Offset: 0x018
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR_CYCLE0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ADDR_CYCLE0[7:0] NAND Flash Array Address Cycle 0
When five address cycles are used, ADDR_CYCLE0 is the first byte written to the NAND Flash (used by the NFC).

16.19.8 NFC Bank Register

Name: HSMC_BANK
Offset: 0x01C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								BANK
Access								R/W
Reset								0

Bit 0 – BANK Bank Identifier

Value	Description
0	Bank 0 is used.
1	Bank 1 is used.

16.19.9 PMECC Configuration Register

Name: HSMC_PMECCFG
Offset: 0x070
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				AUTO				SPAREEN
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
				NANDWR			PAGESIZE[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
				SECTORSZ		BCH_ERR[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit 20 – AUTO Automatic Mode Enable
This bit is only relevant in NAND Read mode, when HSMC_PMECCFG.SPAREEN=0 and HSMC_CFG.RSPARE=1.

Value	Description
0	Indicates that the ECC computation is not retrieved automatically and must be read via User mode.
1	Indicates that the ECC computation is retrieved automatically via Data mode.

Bit 16 – SPAREEN Spare Enable
For NAND write access:
0: The spare area is not protected by ECC.
1: The spare area is protected with the last sector of data.
For NAND read access:
0: The spare area is not protected by ECC.
1: The spare area contains protected data.

Bit 12 – NANDWR NAND Write Access

Value	Description
0	NAND read access.
1	NAND write access.

Bits 9:8 – PAGESIZE[1:0] Number of Sectors in the Page

Value	Name	Description
0	PAGESIZE_1SEC	1 sector for main area (512 or 1024 bytes).
1	PAGESIZE_2SEC	2 sectors for main area (1024 or 2048 bytes).
2	PAGESIZE_4SEC	4 sectors for main area (2048 or 4096 bytes).
3	PAGESIZE_8SEC	8 sectors for main area (4096 or 8192 bytes).

Bit 4 – SECTORSZ Sector Size

Value	Description
0	The ECC computation is based on a sector of 512 bytes.

Value	Description
1	The ECC computation is based on a sector of 1024 bytes.

Bits 2:0 – BCH_ERR[2:0] Error Correcting Capability

Value	Name	Description
0	BCH_ERR2	2 errors
1	BCH_ERR4	4 errors
2	BCH_ERR8	8 errors
3	BCH_ERR12	12 errors
4	BCH_ERR24	24 errors
5	BCH_ERR32	32 errors

16.19.10 PMECC Spare Area Size Register

Name: HSMC_PMECCSAREA
Offset: 0x074
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								SPARESIZE[8]
Access								R/W
Reset								0
Bit	7	6	5	4	3	2	1	0
	SPARESIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – SPARESIZE[8:0] Spare Area Size
Number of bytes in the spare area. The spare area size is equal to (SPARESIZE + 1) bytes.

16.19.11 PMECC Start Address Register

Name: HSMC_PMECCSADDR
Offset: 0x078
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								STARTADDR[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	STARTADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – STARTADDR[8:0] ECC Area Start Address
This register is programmed with the start ECC start address. When STARTADDR is equal to 0, then the first ECC byte is located at the first byte of the spare area.

16.19.12 PMECC End Address Register

Name: HSMC_PMECCEADDR
Offset: 0x07C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								ENDADDR[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	ENDADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – ENDADDR[8:0] ECC Area End Address
This register is programmed with the start ECC end address. When ENDADDR is equal to N, then the first ECC byte is located at byte N of the spare area.

16.19.13 PMECC Control Register

Name: HSMC_PMECCTRL
Offset: 0x084
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DISABLE	ENABLE		USER	DATA	RST
Access			W	W		W	W	W
Reset			–	–		–	–	–

Bit 5 – DISABLE PMECC Enable

Value	Description
0	No effect.
1	Disables the PMECC controller.

Bit 4 – ENABLE PMECC Enable

Value	Description
0	No effect.
1	Enables the PMECC controller.

Bit 2 – USER Start a User Mode Phase

Value	Description
0	No effect.
1	The PMECC controller enters a User mode phase.

Bit 1 – DATA Start a Data Phase

Value	Description
0	No effect.
1	The PMECC controller enters a Data phase.

Bit 0 – RST Reset the PMECC Module

Value	Description
0	No effect.
1	Resets the PMECC controller.

16.19.14 PMECC Status Register

Name: HSMC_PMECCSR
Offset: 0x088
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				ENABLE				BUSY
Access				R				R
Reset				0				0

Bit 4 – ENABLE PMECC Enable Bit

Value	Description
0	PMECC controller disabled.
1	PMECC controller enabled.

Bit 0 – BUSY PMECC Kernel is Busy

Value	Description
0	PMECC controller finite state machine reached idle state.
1	PMECC controller finite state machine is processing the incoming byte stream.

16.19.15 PMECC Interrupt Enable Register

Name: HSMC_PMECCIER
Offset: 0x08C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ERRIE
Access								W
Reset								–

Bit 0 – ERRIE Error Interrupt Enable

Value	Description
0	No effect.
1	The Multibit error interrupt is enabled. An interrupt will be raised if at least one error is detected in at least one sector.

16.19.16 PMECC Interrupt Disable Register

Name: HSMC_PMECCIDR
Offset: 0x090
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ERRID
Access								W
Reset								–

Bit 0 – ERRID Error Interrupt Disable

Value	Description
0	No effect.
1	Multibit error interrupt disabled.

16.19.17 PMECC Interrupt Mask Register

Name: HSMC_PMECCIMR
Offset: 0x094
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ERRIM
Access								R
Reset								0

Bit 0 – ERRIM Error Interrupt Mask

Value	Description
0	Multibit error disabled.
1	Multibit error enabled.

16.19.18 PMECC Interrupt Status Register

Name: HSMC_PMECCISR
Offset: 0x098
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERRIS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ERRIS[7:0] Error Interrupt Status Register
When set to one, bit i of the HSMC_PMECCISR indicates that sector i is corrupted.

16.19.19 PMECC Redundancy x Register

Name: HSMC_PMECCx
Offset: 0xB0 + x*0x04 [x=0..13]
Reset: 0x00000000
Property: Read-only

Note: The block of registers HSMC_PMECCx[x=0..13] is instanced 8 times in the user interface.

Bit	31	30	29	28	27	26	25	24
	ECC[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ECC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ECC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ECC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ECC[31:0] BCH Redundancy
This register contains the remainder of the division of the codeword by the generator polynomial.

16.19.20 PMECC Remainder x Register

Name:	HSMC_REMx
Offset:	0x02B0 + x*0x04 [x=0..15]
Reset:	0x00000000
Property:	Read-only

Note: The block of registers HSMC_REMx[x=0..15] is instanced 8 times in the user interface.

Bit	31	30	29	28	27	26	25	24
			REM2NP3[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	REM2NP3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			REM2NP1[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	REM2NP1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:16 – REM2NP3[13:0] BCH Remainder 2 * N + 3
When sector size is set to 512 bytes, bit REM2NP3[29] is not used and read as zero.
If bit i of the REM2NP3 field is set to one, then the coefficient of the X^i is set to one; otherwise, the coefficient is zero.

Bits 13:0 – REM2NP1[13:0] BCH Remainder 2 * N + 1
When sector size is set to 512 bytes, bit REM2NP1[13] is not used and read as zero.
If bit i of the REM2NP1 field is set to one, then the coefficient of the X^i is set to one; otherwise, the coefficient is zero.

16.19.21 PMECC Error Location Configuration Register

Name: HSMC_ELCFG
Offset: 0x500
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				ERRNUM[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SECTORSZ
Access								R/W
Reset								0

Bits 20:16 – ERRNUM[4:0] Number of Errors

Bit 0 – SECTORSZ Sector Size

Value	Description
0	The ECC computation is based on a 512-byte sector.
1	The ECC computation is based on a 1024-byte sector.

16.19.22 PMECC Error Location Primitive Register

Name: HSMC_ELPRIM
Offset: 0x504
Reset: 0x401A
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PRIMITIV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PRIMITIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	1	0	1	0

Bits 15:0 – PRIMITIV[15:0] Primitive Polynomial
This field indicates the Primitive Polynomial used in the ECC computation.

16.19.23 PMECC Error Location Enable Register

Name: HSMC_ELEN
Offset: 0x508
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			ENINIT[13:8]					
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ENINIT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 13:0 – ENINIT[13:0] Error Location Enable
Initial bit number in the codeword.

16.19.24 PMECC Error Location Disable Register

Name: HSMC_ELDIS
Offset: 0x50C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DIS
Access								W
Reset								–

Bit 0 – DIS Disable Error Location Engine

Value	Description
0	No effect.
1	Disables the Error location engine.

16.19.25 PMECC Error Location Status Register

Name: HSMC_ELSR
Offset: 0x510
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								BUSY
Access								R
Reset								0

Bit 0 – BUSY Error Location Engine Busy

Value	Description
0	Error location engine is disabled.
1	Error location engine is enabled and is finding roots of the polynomial.

16.19.26 PMECC Error Location Interrupt Enable Register

Name: HSMC_ELIER
Offset: 0x514
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								W
Reset								–

Bit 0 – DONE Computation Terminated Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt.

16.19.27 PMECC Error Location Interrupt Disable Register

Name: HSMC_ELIDR
Offset: 0x518
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								W
Reset								–

Bit 0 – DONE Computation Terminated Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt.

16.19.28 PMECC Error Location Interrupt Mask Register

Name: HSMC_ELIMR
Offset: 0x51C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R
Reset								0

Bit 0 – DONE Computation Terminated Interrupt Mask

Value	Description
0	Computation Terminated interrupt disabled.
1	Computation Terminated interrupt enabled.

16.19.29 PMECC Error Location Interrupt Status Register

Name: HSMC_ELISR
Offset: 0x520
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			ERR_CNT[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R
Reset								0

Bits 13:8 – ERR_CNT[5:0] Error Counter value
This field indicates the number of roots of the polynomial.

Bit 0 – DONE Computation Terminated Interrupt Status
When set to one, this indicates that the error location engine has completed the root finding algorithm.

16.19.30 PMECC Error Location SIGMA0 Register

Name: HSMC_SIGMA0
Offset: 0x528
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			SIGMA0[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGMA0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 13:0 – SIGMA0[13:0] Coefficient of degree 0 in the SIGMA polynomial
SIGMA0 belongs to the finite field GF(2^{13}) when the sector size is set to 512 bytes.
SIGMA0 belongs to the finite field GF(2^{14}) when the sector size is set to 1024 bytes.

16.19.31 PMECC Error Location SIGMAx Register

Name: HSMC_SIGMAx
Offset: 0x052C + (x-1)*0x04 [x=1..32]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			SIGMAx[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGMAx[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – SIGMAx[13:0] Coefficient of degree x in the SIGMA polynomial
SIGMAx belongs to the finite field GF(2^{13}) when the sector size is set to 512 bytes.
SIGMAx belongs to the finite field GF(2^{14}) when the sector size is set to 1024 bytes.

16.19.32 PMECC Error Location x Register

Name: HSMC_ERRLOCx
Offset: 0x05AC + x*0x04 [x=0..31]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
			ERRLOCN[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	ERRLOCN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – ERRLOCN[13:0] Error Position within the Set {sector area, spare area}
ERRLOCN points to 1 when the first bit of the main area is corrupted.
If the sector size is set to 512 bytes, the ERRLOCN points to 4096 when the last bit of the sector area is corrupted.
If the sector size is set to 1024 bytes, the ERRLOCN points to 8192 when the last bit of the sector area is corrupted.
If the sector size is set to 512 bytes, the ERRLOCN points to 4097 when the first bit of the spare area is corrupted.
If the sector size is set to 1024 bytes, the ERRLOCN points to 8193 when the first bit of the spare area is corrupted.

16.19.33 Setup Register

Name: HSMC_SETUPx
Offset: 0x0700 + x*0x14 [x=0..3]
Reset: 0x01010101
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			NCS_RD_SETUP[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
			NRD_SETUP[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
			NCS_WR_SETUP[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
			NWE_SETUP[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	1

Bits 29:24 – NCS_RD_SETUP[5:0] NCS Setup Length in Read Access
In Read access, the NCS signal setup length is defined as:
NCS setup length = (128 * NCS_RD_SETUP[5] + NCS_RD_SETUP[4:0]) clock cycles.

Bits 21:16 – NRD_SETUP[5:0] NRD Setup Length
The NRD signal setup length is defined as:
NRD setup length = (128 * NRD_SETUP[5] + NRD_SETUP[4:0]) clock cycles.

Bits 13:8 – NCS_WR_SETUP[5:0] NCS Setup Length in Write Access
In write access, the NCS signal setup length is defined as:
NCS setup length = (128 * NCS_WR_SETUP[5] + NCS_WR_SETUP[4:0]) clock cycles.

Bits 5:0 – NWE_SETUP[5:0] NWE Setup Length
The NWE signal setup length is defined as:
NWE setup length = (128 * NWE_SETUP[5] + NWE_SETUP[4:0]) clock cycles.

16.19.34 Pulse Register

Name: HSMC_PULSEx
Offset: 0x0704 + x*0x14 [x=0..3]
Reset: 0x01010101
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		NCS_RD_PULSE[6:0]						
Access		W	W	W	W	W	W	W
Reset		0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
		NRD_PULSE[6:0]						
Access		W	W	W	W	W	W	W
Reset		0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
		NCS_WR_PULSE[6:0]						
Access		W	W	W	W	W	W	W
Reset		0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
		NWE_PULSE[6:0]						
Access		W	W	W	W	W	W	W
Reset		0	0	0	0	0	0	1

Bits 30:24 – NCS_RD_PULSE[6:0] NCS Pulse Length in READ Access
In READ mode, The NCS signal pulse length is defined as:
NCS pulse length = (256 * NCS_RD_PULSE[6] + NCS_RD_PULSE[5:0]) clock cycles.

Bits 22:16 – NRD_PULSE[6:0] NRD Pulse Length
The NRD signal pulse length is defined as:
NRD pulse length = (256 * NRD_PULSE[6] + NRD_PULSE[5:0]) clock cycles.
The NRD pulse width must be as least 1 clock cycle.

Bits 14:8 – NCS_WR_PULSE[6:0] NCS Pulse Length in WRITE Access
In Write access, The NCS signal pulse length is defined as:
NCS pulse length = (256 * NCS_WR_PULSE[6] + NCS_WR_PULSE[5:0]) clock cycles.
The NCS pulse must be at least one clock cycle.

Bits 6:0 – NWE_PULSE[6:0] NWE Pulse Length
The NWE signal pulse length is defined as:
NWE pulse length = (256 * NWE_PULSE[6]+NWE_PULSE[5:0]) clock cycles.
The NWE pulse must be at least one clock cycle.

16.19.35 Cycle Register

Name: HSMC_CYCLEx
Offset: 0x0708 + x*0x14 [x=0..3]
Reset: 0x00030003
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								NRD_CYCLE[8]
Access								R/W
Reset								0
Bit	23	22	21	20	19	18	17	16
	NRD_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8
								NWE_CYCLE[8]
Access								R/W
Reset								0
Bit	7	6	5	4	3	2	1	0
	NWE_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bits 24:16 – NRD_CYCLE[8:0] Total Read Cycle Length
The total read cycle length is the total duration in clock cycles of the read cycle. It is equal to the sum of the setup, pulse and hold steps of the NRD and NCS signals. It is defined as:
Read cycle length = (NRD_CYCLE[8:7] * 256) + NRD_CYCLE[6:0] clock cycles.

Bits 8:0 – NWE_CYCLE[8:0] Total Write Cycle Length
The total write cycle length is the total duration in clock cycles of the write cycle. It is equal to the sum of the setup, pulse and hold steps of the NWE and NCS signals. It is defined as:
Write cycle length = (NWE_CYCLE[8:7] * 256) + NWE_CYCLE[6:0] clock cycles.

16.19.36 Timings Register

Name: HSMC_TIMINGSx
Offset: 0x070C + x*0x14 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	NFSEL				TWB[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TRR[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				OCMS	TAR[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TADL[3:0]				TCLR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – NFSEL NAND Flash Selection
If this bit is set to one, the chip select is assigned to NAND Flash write enable and read enable lines drive the Error Correcting Code module.

Bits 27:24 – TWB[3:0] WEN High to REN to Busy
Write Enable rising edge to Ready/Busy falling edge timing.
Write Enable to Read/Busy = (TWB[3] * 64) + TWB[2:0] clock cycles.

Bits 19:16 – TRR[3:0] Ready to REN Low Delay
Ready/Busy signal to Read Enable falling edge timing.
Read to REN = (TRR[3] * 64) + TRR[2:0] clock cycles.

Bit 12 – OCMS Off Chip Memory Scrambling Enable
When set to one, the memory scrambling is activated. (Value must be zero if external memory is NAND Flash and NFC is used).

Bits 11:8 – TAR[3:0] ALE to REN Low Delay
Address Latch Enable falling edge to Read Enable falling edge timing.
Address Latch Enable to Read Enable = (TAR[3] * 64) + TAR[2:0] clock cycles.

Bits 7:4 – TADL[3:0] ALE to Data Start
Last address latch cycle to the first rising edge of WEN for data input.
Last address latch to first rising edge of WEN = (TADL[3] * 64) + TADL[2:0] clock cycles.

Bits 3:0 – TCLR[3:0] CLE to REN Low Delay
Command Latch Enable falling edge to Read Enable falling edge timing.
Latch Enable Falling to Read Enable Falling = (TCLR[3] * 64) + TCLR[2:0] clock cycles.

16.19.37 Mode Register

Name: HSMC_MODEx
Offset: 0x0710 + x*0x14 [x=0..3]
Reset: 0x00001003
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DBW				BAT
Access				R/W				R/W
Reset				1				0

Bit	7	6	5	4	3	2	1	0
			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
Access			R/W	R/W			R/W	R/W
Reset			0	0			1	1

Bit 12 – DBW Data Bus Width

Value	Name	Description
0	BIT_8	8-bit bus
1	BIT_16	16-bit bus

Bit 8 – BAT Byte Access Type
This field is used only if DBW defines a 16-bit data bus.

Value	Name	Description
0	BYTE_SELECT	Byte select access type: • Write operation is controlled using NCS, NWE, NBS0, NBS1. • Read operation is controlled using NCS, NRD, NBS0, NBS1.
1	BYTE_WRITE	Byte write access type: • Write operation is controlled using NCS, NWR0, NWR1. • Read operation is controlled using NCS and NRD.

Bits 5:4 – EXNW_MODE[1:0] NWAIT Mode
The NWAIT signal is used to extend the current read or write signal. It is only taken into account during the pulse phase Read and Write controlling signal. When the use of NWAIT is enabled, at least one cycle hold duration must be programmed for the read and write controlling signal.

Value	Name	Description
0	DISABLED	Disabled—The NWAIT input signal is ignored on the corresponding Chip Select.
1	–	Reserved
2	FROZEN	Frozen Mode—If asserted, the NWAIT signal freezes the current read or write cycle. After deassertion, the read/write cycle is resumed from the point where it was stopped.
3	READY	Ready Mode—The NWAIT signal indicates the availability of the external device at the end of the pulse of the controlling read or write signal, to complete the access. If high, the access normally completes. If low, the access is extended until NWAIT returns high.

Bit 1 – WRITE_MODE Selection of the Control Signal for Write Operation

Value	Name	Description
0	NCS_CTRL	The Write operation is controller by the NCS signal.
1	NWE_CTRL	The Write operation is controlled by the NWE signal

Bit 0 – READ_MODE Selection of the Control Signal for Read Operation

Value	Name	Description
0	NCS_CTRL	The Read operation is controlled by the NCS signal.
1	NRD_CTRL	The Read operation is controlled by the NRD signal.

16.19.38 Off Chip Memory Scrambling Register

Name: HSMC_OCMS
Offset: 0x7A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SRSE	SMSE
Access							R/W	R/W
Reset							0	0

Bit 1 – SRSE NFC Internal SRAM Scrambling Enable

Value	Description
0	Disable Scrambling for NFC internal SRAM access.
1	Enable Scrambling for NFC internal SRAM access. (OCMS bit must be cleared in the corresponding HSMC_TIMINGSx register.)

Bit 0 – SMSE Static Memory Controller Scrambling Enable

Value	Description
0	Disable “Off Chip” Scrambling for SMC access.
1	Enable “Off Chip” Scrambling for SMC access. (If OCMS bit is set in the corresponding HSMC_TIMINGSx register.)

16.19.39 Off Chip Memory Scrambling Key1 Register

Name: HSMC_KEY1
Offset: 0x7A4
Reset: 0x00000000
Property: Write-once

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
Access	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	KEY1[23:16]							
Access	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	KEY1[15:8]							
Access	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	KEY1[7:0]							
Access	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once	W-Once
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY1[31:0] Off Chip Memory Scrambling (OCMS) Key Part 1
When Off Chip Memory Scrambling is enabled by setting the HSMC_OCMS and HSMC_TIMINGS registers in accordance, the data scrambling depends on KEY1 and KEY2 values.

16.19.40 Off Chip Memory Scrambling Key2 Register

Name: HSMC_KEY2
Offset: 0x7A8
Reset: 0x00000000
Property: Write-once

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access	W-once	W-once	W-once	W-once	W-once	W-once	W-once	W-once
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access	W-once	W-once	W-once	W-once	W-once	W-once	W-once	W-once
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access	W-once	W-once	W-once	W-once	W-once	W-once	W-once	W-once
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access	W-once	W-once	W-once	W-once	W-once	W-once	W-once	W-once
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY2[31:0] Off Chip Memory Scrambling (OCMS) Key Part 2
When Off Chip Memory Scrambling is enabled by setting the HSMC_OCMS and HSMC_TIMINGS registers in accordance, the data scrambling depends on KEY2 and KEY1 values.

16.19.41 Clock Configuration Register

Name: HSMC_CLKCFG
Offset: 0x7AC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
							CLKEDGE	CLKEN
Access							R/W	R/W
Reset							0	0
Bit	15	14	13	12	11	10	9	8
								CLKDIV[8]
Access								R/W
Reset								0
Bit	7	6	5	4	3	2	1	0
	CLKDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – CLKEDGE SMC Clock Edge

Value	Description
0	The start of the external bus access is aligned with the rising edge of the SMC clock output.
1	The start of the external bus access is aligned with the falling edge of the SMC clock output.

Bit 16 – CLKEN SMC Clock Enable

Value	Description
0	The SMC clock is disabled.
1	The SMC clock is enabled.

Bits 8:0 – CLKDIV[8:0] SMC Clock Divider
The SMC clock output frequency is equal to periph_clk/(CLKDIV + 1). If CLKDIV = 0, the SMC clock output frequency is equal to periph_clk/2.

16.19.42 Write Protection Mode Register

Name: HSMC_WPMR
Offset: 0x7E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534D43	PASSWD	Writing any other value in this field aborts the write operation of bit WPEN. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for list of write-protected registers.

Value	Description
0	Disables write protection if WPKEY value corresponds to 0x534D43 (“SMC” in ASCII)
1	Enables write protection if WPKEY value corresponds to 0x534D43 (“SMC” in ASCII)

16.19.43 Write Protection Status Register

Name: HSMC_WPSR
Offset: 0x7E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protect violation has occurred since the last read of the HSMC_WPSR.
1	A write protect violation has occurred since the last read of the HSMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

17. Universal DDR Memory Controller (UDDRC)

17.1 Description

The SDRAM memory interface is made of:

- the Universal DDR Memory Controller (UDDRC) and
- the PHYsical layer interface (DDR3PHY).

The UDDRC receives transactions from the internal buses. These transactions are queued internally and scheduled for access in order to the DDR-SDRAM while satisfying the DDR-SDRAM protocol timing requirements, transaction priorities, and dependencies between transactions.

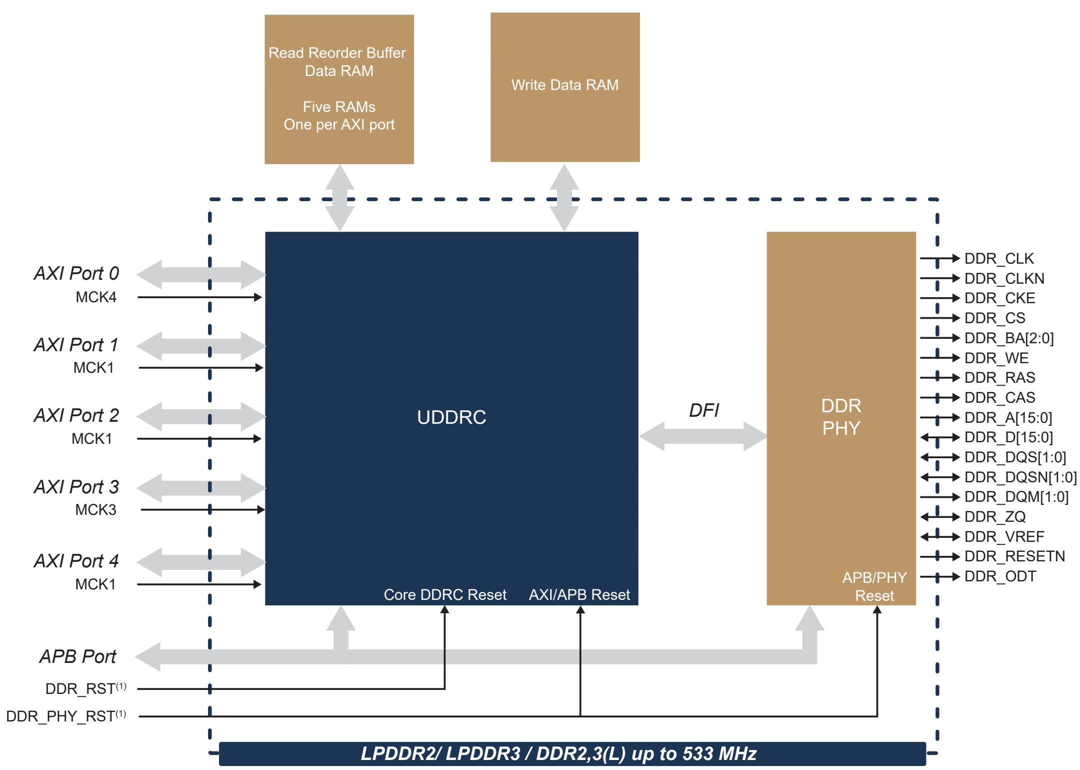
17.2 Embedded Characteristics

- Interface to DDR2, DDR3, LPDDR2, LPDDR3 Memories
- All Signals Supported on Control, Write Data and Read Data Interfaces
- Internal 5 AXI Ports Controller with Arbitration
- External 16-bit DDR Interface
- Support for 1 Memory Rank
- For DDR3, LPDDR2 and LPDDR3 Configurations: Direct Software Request Control or Programmable Internal Control for ZQ Short Calibration Cycles
- For DDR3, LPDDR2 and LPDDR3 Configurations: Support for ZQ Long Calibration after Self-refresh Exit
- For LPDDR2 and LPDDR3 Configurations: Support for ZQ Reset Feature through Software
- Dynamic Scheduling to Optimize Bandwidth and Latency
- Read and Write Buffers in Fully Associative CAMs (Content Addressable Memories)
- Delayed Writes for Optimum Performance on SDRAM Data Bus
- Software-programmable Quality of Service (QoS) Support
- Control Options to Avoid Starvation of Lower Priorities
- Coherency for Write-After-Read (WAR) and Read-After-Write (RAW) Hazards
- Paging Policy Selectable by Configuration Registers:
  - Leave pages open after accesses, or
  - Close page when no further accesses are available in the controller for that page, or
  - Auto-precharge with each access, with an optimization for Page Close mode which leaves the page open after a flush for read-write and write-read collision cases
- Supports Automatic DDR-SDRAM Power-down Entry and Exit caused by Lack of Incoming Transaction for a Programmable Time
- Supports Automatic Clock Stop (LPDDR2/LPDDR3) Entry and Exit caused by Lack of Incoming Transaction
- Supports Automatic UDDRC Low-power Mode Operation caused by Lack of Transaction Arrival for a Programmable Time through the Hardware Low-power Interface
- Supports Self-refresh Entry and Exit as follows:
  - Support for automatic self-refresh entry and exit caused by lack of incoming transaction for a programmable time
  - Support for self-refresh entry and exit under software control
  - Support for self-refresh entry and exit using dedicated DDRC hardware low-power interface control
- Support for Dynamically Changing Clock Frequency while in Self-refresh:
  - DDR3 DLL-off mode supported
  - Shadow timing registers provided to allow fast frequency changing
- Support for Deep Power-down Entry and Exit under Software Control (LPDDR2/LPDDR3)
- Support for explicit SDRAM Mode Register Updates under Software Control
- Flexible Address Mapper Logic to allow Application-specific Mapping of Row, Column, Bank, and Rank Bits

- Programmable Support for 1T or 2T Timing
- User-selectable Refresh Control Options:
  – Controller-generated auto-refreshes at programmable average intervals
  – Ability to disable controller-generated auto-refreshes
  – Ability to issue a refresh through direct software request
  – When LPDDR2/LPDDR3 is used, user-selectable ability to perform per-bank refreshes rather than all-banks refreshes
- Advanced Power-saving Design Includes No Unnecessary Toggling of Command, Address, and Data Pins (RAS/CAS/WE/BA/A hold last state after each command; DDR_D does not transition on writes when bytes are disabled)
- Support for DDR2, DDR3, DDR3L, LPDDR2, LPDDR3
- Leverages Out-of-order Requests to Maximize Throughout
- The UDDRC Implements the Following Standards:
  – JEDEC DDR2 SDRAM Specification, JESD79-2
  – JEDEC DDR3 SDRAM Specification, JESD79-3
  – JEDEC LPDDR2 SDRAM Specification, JESD209-2
  – JEDEC LPDDR3 SDRAM Specification, JESD209-3

17.3 Block Diagram

Figure 17-1. UDDRC Block Diagram



(1) DDR_RST and DDR_PHY_RST are driven by the reset controller. Refer to the “Reset Controller (RSTC)” section for more details.

17.4 I/O Lines Description

Table 17-1. DDR/LPDDR I/O Lines Description

Name	Function	Type	Active Level
DDR/LPDDR Controller			
VDDIODDR	Power Supply of memory interface	Power	–
DDR_VREF	Reference Voltage	Input	–
DDR_ZQ	Calibration reference	Input	–
DDR_ODT	On-Die-Termination	Output	–
DDR_D[15:0]	Data Bus	I/O	–
DDR_A[15:0]	Address Bus	Output	–
DDR_DQM[1:0]	Data Mask	Output	–
DDR_DQS[1:0]	Data Strobe	I/O	–
DDR_DQSN[1:0]	Negative Data Strobe	I/O	–
DDR_CS	Chip Select	Output	Low
DDR_RESETN	DDR3 Active Low Asynchronous Reset	Output	Low
DDR_CLK, DDR_CLKN	Differential Clock	Output	–
DDR_CKE	Clock enable	Output	High
DDR_RAS	Row signal	Output	Low
DDR_CAS	Column signal	Output	Low
DDR_WE	Write enable	Output	Low
DDR_BA[2:0]	Bank Select	Output	–

17.4.1 Product Dependencies

The table below gives the connections to the various memory types.

Table 17-2. I/O Lines Usage vs. Operating Modes

Signal Name	DDR2	DDR3	DDR3L	LPDDR2/LPDDR3
DDR_VREF	VDDIODDR/2	VDDIODDR/2	VDDIODDR/2	VDDIODDR/2
DDR_ZQ	GND via 240 Ω	GND via 240 Ω	GND via 240 Ω	GND via 240 Ω
DDR_CK, DDR_CLKN	CLK, CLKN	CLK, CLKN	CLK, CLKN	CLK, CLKN
DDR_CKE	CLKE	CLKE	CLKE	CLKE
DDR_CS	CS	CS	CS	CS
DDR_RESETN	Not connected	DDR_RESETN	DDR_RESETN	Not connected
DDR_BA[2:0]	BA[2:0]	BA[2:0]	BA[2:0]	Not connected
DDR_WE	WE	WE	WE	Not connected
DDR_RAS, DDR_CAS	RAS, CAS	RAS, CAS	RAS, CAS	Not connected
DDR_A[15:0]	A[15:0]	A[15:0]	A[15:0]	CA[9:0]

...........continued				
Signal Name	**DDR2**	**DDR3**	**DDR3L**	**LPDDR2/LPDDR3**
DDR_D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]
DDR_DQS[1:0], DDR_DQSN[1:0]	LDQS,UDQS, DDR_VREF	DQS[1:0], DQSN[1:0]	DQS[1:0], DQSN[1:0]	DQS[1:0], DQSN[1:0]
DDR_DQM[1:0]	UDM, LDM	DQM[1:0]	DQM[1:0]	DQM[1:0]

17.4.2 Implementation Example

Table 17-3. CAx LPDDR2 Signal Connections

DDR Controller Signal	LPDDR2 Signal
DDR_A0	CA0
DDR_A1	CA1
DDR_A2	CA2
DDR_A3	CA3
DDR_A4	CA4
DDR_A5	CA5
DDR_A6	CA6
DDR_A7	CA7
DDR_A8	CA8
DDR_A9	CA9
Higher addresses	Higher CAs

17.5 Functional Description

The UDDRC converts system bus transactions into memory commands that are compliant with the DDR protocols.

The UDDRC performs the following functions:

- Accepts requests from the SoC core with system addresses and associated data for writes.
- Performs address mapping from system addresses to SDRAM addresses (rank, bank, bank group, row).
- Prioritizes requests to minimize the latency of reads (especially high priority reads) and maximize page hits.
- Ensures that the SDRAM is properly initialized.
- Ensures that all requests made to the SDRAM are legal (accounting for associated SDRAM constraints)
- Ensures that refreshes and other SDRAM maintenance requests are inserted as required.
- Controls when the SDRAM enters and exits the various power-saving modes appropriately.

17.5.1 AXI Port Interface (XPI)

The AXI protocol is burst-based with read and write request channels that specify the host ID for the request, start byte address, burst length, burst size, and burst type. This information is processed by the interface and is used subsequently by the UDDRC.

The XPI interfaces the AXI application port to the UDDRC and performs the following main functions:

- Read address generation
- Write address generation
- Write data generation
- Read data and response generation

- Write response generation

The XPI converts AXI bursts into read and write requests, which are forwarded to the Port Arbiter (PA). In the opposite direction, the XPI converts the responses from the DDRC into appropriate AXI responses.

The AXI Specification requires that transactions must not cross a 4K address boundary.

17.5.1.1 Read Address Channel

The AXI read address channel has the following features:

- The read transaction can be of any length up to 256 for incremental bursts, 16 for wrapping bursts.
- The burst types supported are incremental and wrapping.
- The burst start address can be unaligned to the AXI data width boundaries.
- The size of the burst can be less than the full width of the AXI data bus (also known as subsized transfers).

The signals on the read address channel are registered into the XPI in accordance with the AXI valid/read handshaking protocol and are synchronous with the AXI clock (aclk).

Read requests are accepted if the request can be written into the Read Address Queue (RAQ). This is used to store all the read address requests. There is a single address queue on a given port.

Each AXI port can store in its RAQ up to:

- Port 0: 8 addresses
- Port 1: 2 addresses
- Port 2: 16 addresses
- Port 3: 2 addresses
- Port 4: 16 addresses

17.5.1.2 Write Address Channel

Write Address Queue (WAQ) is used to store all the addresses for write requests from a given port. There is a single queue for all AXI IDs from a given port. The write address channel behavior is similar to the read address channel.

Each AXI port can stored in its WAQ up to:

- Port 0: 8 addresses
- Port 1: 2 addresses
- Port 2: 8 addresses
- Port 3: 4 addresses
- Port 4: 16 addresses

Write address and read address channels are independent and the ordering between the write and read requests may not be preserved.

To preserve the sequence, a higher-level protocol needs to wait for read/write response before sending the next transaction.

Transactions across the ports are independent and can be issued in any order.

17.5.1.3 Wrap Burst Expansion

A WRAP burst may be expanded by the XPI into multiple SDRAM transactions. This is also true for cases where WRAP bursts, which could be accommodated by a single transaction on the SDRAM interface.

17.5.1.4 Software Coherency for AXI Ports

The Address Collision Handling section describes how the DDRC handles in-order execution of commands to the same address.

For the commands issued at the AXI port, the logic in the DDRC protects against all types of software coherency hazards when the AXI host waits for write (read) response before sending the next same address read (same address write) or vice-versa.

17.5.2 Port Arbiter (PA)

The Port Arbiter block arbitrates command requests from the AXI ports to the HIF of the DDR Controller (DDRC). PA is comprised of multiple tiers of arbitration stages which include:

- Read/write arbitration
- 2-priority level arbitration based on port aging and expired-VPR/expired-VPW commands (timeout-priority0)
- 2-priority level arbitration for read requests based on DDRC read priorities
- 32-priority level arbitration based on internal port aging or 16-priority level arbitration based external AXI QoS inputs
- Round-robin arbitration to resolve ports having the same priority after passing all stages of arbitration

17.5.2.1 Registers Related to Port Arbiter

The following are the registers related to the Port Arbiter:

- UDDRC_SCHED
- UDDRC_SCHED1
- UDDRC_SCHED2

For more information about these registers, see Register Descriptions.

17.5.3 Address Mapper

Read and write requests are provided to the UDDRC with a system address. The system address is the command address of a transaction as presented on one of the data ports. The UDDRC converts this system address to a physical address. It maps the system address to the SDRAM rank, bank, row, and column addresses.

The first part of the mapping is conversion of system address to AXI byte address. The controller maps disjoint address regions to internal consecutive addresses. Otherwise, the controller assumes that the DRAM is always mapped as a monolithic block. For more information about this, see System Address Regions.

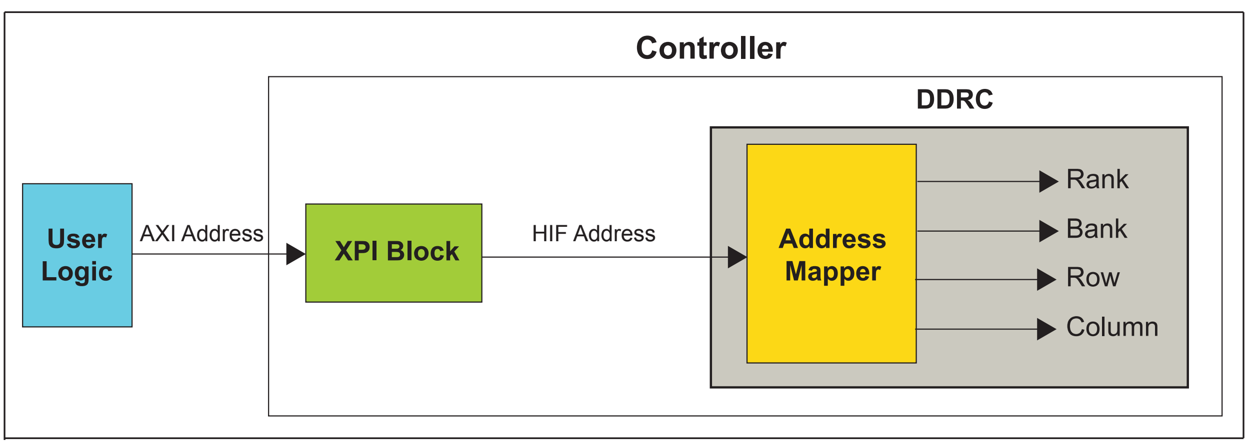
The second part of this mapping is the conversion of AXI byte address to HIF word address. This is performed in the XPI block, and is applicable to AXI configurations only. For more information about this, see Application to HIF Address Mapping.

The last part is the conversion of HIF word address to SDRAM address. A flexible address mapper maps the HIF word address to the SDRAM rank/bank/bank group/row/column address. This address mapper is located within the DDRC.

The address mapping to be used depends on the use-case. The UDDRC provides a set of registers that allows flexible re-programming of logical to physical address mapping. For more information about this, see HIF Address to SDRAM Address Mapping.

The following figure explains the conversion of AXI to HIF to the SDRAM rank/bank/row/column/bank group addresses.

Figure 17-2. Conversion of AXI to HIF to SDRAM Rank/Bank/Row/Column/Bank Group Addresses



17.5.3.1 System Address Regions

The system address regions add the capability to define up to four disjoint memory regions mapping to the DRAM as consecutive addresses.

Each region is defined by:

- Base Address: starting address of the region aligned to the minimum block size.
- Number of Blocks: size of the region in multiples of minimum block size.

The base address of each region is specified by the register SARBASEn.base_addr and the total number of blocks is specified by the register SARSIZEn.nblocks (see Register Descriptions).

System address region specification is the same for all ports. Error response is not generated by the controller for addresses falling outside the specified address regions and the same address translation is applied from one base address to the next base address.

The base addresses must be specified such that they are in the ascending order (SARBASE0 < SARBASE1 < SARBASE2 < SARBASE3) and such that regions do not overlap.

17.5.3.2 Application to HIF Address Mapping

The {ARADDR | AWADDR} is a byte address. XPI maps the MSB bits of the application address to the HIF address (hif_cmd_addr) in the following ways:

- hif_cmd_addr [36:3] = {ARADDR | AWADDR} [31:4]
- hif_cmd_addr[2] is internally generated by XPI
- hif_cmd_addr [1:0] = 0

17.5.3.3 HIF Address to SDRAM Address Mapping

The address mapper maps HIF word addresses to SDRAM addresses by selecting the HIF address bit that maps to each and every applicable SDRAM address bit. While it is possible to map HIF address bits to an SDRAM address in any desired manner, the full available address space is accessible only when no two SDRAM address bits are determined by the same HIF address bit. Each SDRAM address bit has an associated register vector to determine its source.

Registers ADDRMAPx (x = 0 to 11) are used to program the address mapper. For more information on ADDRMAP registers, see Register Descriptions.

The HIF address bit number is determined by adding the internal base of the ADDRMAPx (x = 0 to 11) register to the programmed value for that register, as described in the following equation:

HIF address bit number = [internal base] + [register value]

For example, for ADDRMAP3.addrmap_col_b7, the internal base is 7. When full data bus is in use, column bit 7 is determined by the following equation:

7 + [register value]

If this register is programmed to 2, the HIF address bit can be calculated by using following equation:

[HIF address bit number] = 7 + 2 = 9

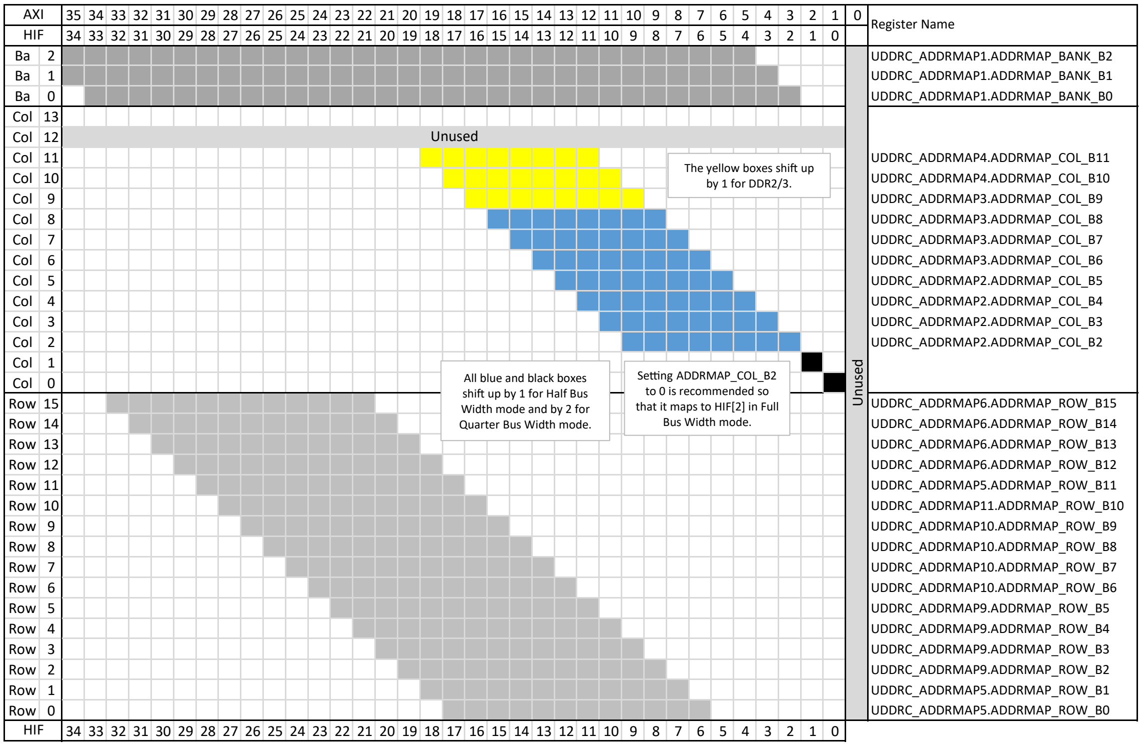
In other words, the column address bit 7 sent to SDRAM would always be equal to hif_cmd_addr[9] of the corresponding HIF source address.

Notes:

1. All of the column bits shift up 1 bit when only half of the data bus is in use. In this case, you need to look at ADDRMAP3.addrmap_col_b6 instead to determine the value of column address bit 7.
2. All of the column bits shift up 2 bits when only a quarter of the data bus is in use. In this case, you need to look at ADDRMAP2.addrmap_col_b5 instead to determine the value of column address bit 7.
3. The addressing for column bits is independent of hardware configuration. The column bits do not shift up by an additional 1 bit.
4. The register ADDRMAP5.addrmap_row_b2_10 can map multiple HIF address bits (if the value set is lesser than 15). If set to 15, then ADDRMAP9, ADDRMAP10, and ADDRMAP11 registers are used, which provide individual mapping for the HIF address bits (rows 2 to 10).
5. For any address bits which cannot be in use in all cases, all bits of the associated address map register must be set to 1 when the associated SDRAM address bit is not in use.

The system address to physical address mapping can be done by choosing any one of the possible combinations from the following figure. It explains the different possible ways to map the HIF address bits to the SDRAM rank/bank/bank group/row/column address.

Figure 17-3. System Address to Physical Address Mapping



17.5.3.4 Registers Related to Address Mapper

The following are the registers related to the Address Mapper:

- UDDRC_ADDRMAP1
- UDDRC_ADDRMAP2
- UDDRC_ADDRMAP3
- UDDRC_ADDRMAP4
- UDDRC_ADDRMAP5
- UDDRC_ADDRMAP6
- UDDRC_ADDRMAP9
- UDDRC_ADDRMAP10
- UDDRC_ADDRMAP11

For more information about these registers, see Register Descriptions.

17.5.4 Address Collision Handling

The UDDRC can execute transactions out-of-order while ensuring that all transactions appear as if they are executed in the order in which they are received. Every transaction that requires a response from the UDDRC arrives with a token number which is provided back to the SoC core as part of the response. Since the UDDRC queues transactions prior to execution, it is possible that multiple transactions to the same SDRAM address arrive before the first transaction to that address is issued.

For address collision, two HIF addresses are considered the same address if all of the HIF address bits (except for the LSB column values) are the same. That is, the "collision" ignores the LSB column values when comparing addresses. The following HIF address bits are ignored during comparison:

- HIF[2+addrmap_col_b2]
- HIF[1:0]

17.5.5 Quality of Service (QoS)

Quality of Service is extremely important in distributing the available resources (bandwidth or latency) amongst the different devices in a fair mechanism to meet the overall performance requirements.

In the circuit, there are multiple hosts trying to communicate with different clients through Interconnect. Interconnect is used to route the traffic between the hosts and clients. There are potential chances of bandwidth sensitive host starving for data and latency sensitive host requirements are not met. This impacts the overall performance of the subsystem. QoS addresses these kinds of performance issues in subsystem.

17.5.5.1 Traffic Classes

The UDDRC supports different traffic classes that are distinguished from each other by the internal ‘arqos’ signal.

17.5.5.1.1 Read Classes

- Low Priority Read (LPR): also called “the best effort traffic”. There is no resource allocation and it shares the resources with the other traffic types. LPR is always treated as low priority in the PA and in the DDRC. There are timeout mechanisms that can be used to prevent starvation for both the PA and the DDRC. When there is a timeout in the PA, that port becomes the highest priority (priority0). When there is a timeout in the LPR store in the DDRC (in other words, LPR store becomes critical), the LPR entries are served before HPR entries.
- Variable Priority Read (VPR): also called “the maximum latency bound traffic” for meeting real time deadlines in video or audio applications. VPR traffic shares the same resources with LPR in the DDRC. But, based on the configuration, it may have a dedicated queue (red or blue) in the XPI. It has the same priority as LPR, but lower priority than HPR for the PA. For the DDRC, VPR has the same initial priority as LPR. Each command tagged as VPR has an associated latency timer. When expired, VPR transactions have the highest priority in the controller, both in the PA and in the DDRC. The purpose of this traffic class is to limit the maximum latency, where this latency bound is expected to be a few hundred clock cycles.
- High Priority Read (HPR): it has allocated resources. If the XPIs are configured to have dual read address queues, then the red queue can be allocated to HPR. For DDRC, HPR traffic goes to its own dedicated store. HPR traffic has higher priority than LPR. HPR is meant for latency critical but not real time applications such as CPU.

17.5.5.1.2 Write Classes

- Normal Priority Write (NPW): it is also called as the best effort traffic. There is no resource allocation and it shares the resources with the other traffic types. It is always treated as normal priority in the PA and DDRC. There are timeout mechanisms that can be used to prevent starvation for the PA. When there is a timeout in the PA, that port becomes the highest priority (priority0).
- Variable Priority Write (VPW): it is also called as the maximum latency bound traffic to meet real time deadlines in video or audio applications. VPW traffic shares the same resources with NPW in the DDRC. For the DDRC, VPW has the same initial priority as NPW. Each command tagged as VPW has an associated latency timer. When expired, VPW transactions have the highest priority in the controller, both in the PA and in the DDRC. The purpose of this traffic class is to limit the maximum latency, where this latency bound is expected to be a few hundred clock cycles.

17.5.5.1.3 QoS Mapping

The priority of an initiator (host) is susceptible to change by the intermediate agents (for example, interconnect) before reaching the destination client depending on the service levels provided with respect to the required targets. Therefore, the relationship between the IDs and their class representations must be dynamically determined.

Arqos/awqos internal signal determines both port priorities and DDRC priorities dynamically. 16 QoS levels are divided to three regions for reads, two regions for writes. Each region can be assigned to any of the following traffic class—LPR, VPR, and HPR for reads, NPW and VPW for writes.

Regions 0 and 1 are assigned to the blue address queue (see Dual Read Address Queue). Typically, LPR/NPW and VPR/VPW may share this resource.

Region 2 is assigned to the red queue (see Dual Read Address Queue) for reads (when dual queue is selected), while for writes it is assigned to the same queue as Region 0 and Region 1 (being writes, always single queue). Region 2 can be mapped to HPR or VPR traffic for reads, VPW or NPW traffic for writes. In the DDRC, LPR/NPW and VPR/VPW share the same CAM store called LPR/NPW store. HPR has its own store in the DDRC.

The regions are mapped per port using PCFGQOS0_n and PCFGWQOS0_n registers (see Register Descriptions).

Fields for PCFGQOS0_n register are as follows:

- rqos_map_level<x> (where x is 1 to 2) indicates two separation levels for three regions. Possible values 0 to 14 correspond to an arqos value. These registers indicate the upper end of the region. For example, if rqos_map_level2 is set to 14, then Region 2 is identified as all transactions with arqos value of 15.
- rqos_map_region<y> (where y is 0 to 2) for region identifier. This register indicates the traffic class of each of the three regions. rqos_map_region2 is present only in dual read queue configurations (see Dual Read Address Queue).
  Valid values are:
  - 0–LPR
  - 1–VPR
  - 2–HPR

For single address queue configurations, Region 0 and Region 1 map to the blue queue. Being single queue, Region 2 is not present. In this case registers rqos_map_level2 and rqos_map_region2 do not exist. In this case Region 0 can be set to HPR or VPR, Region 1 to VPR or LPR.Fields for PCFGWQOS0_n register are as follows:

- wqos_map_level<x> (where x is 1 to 2) indicates the separation level for three regions Possible values 0 to 14 correspond to an awqos value. These registers indicate the upper end of the region. For example, if wqos_map_level2 is set to 14, then Region 2 is identified as all transactions with awqos value of 15.
- wqos_map_region<y> (where y is 0 to 2) for region identifier. This register indicates the traffic class of each of the two regions.

Valid values are:

- 0–NPW
- 1–VPW

Writes are always single queue: Region 0, Region 1 and Region 2 map all to the same queue. All fields always exist.

17.5.5.1.4 Dual Read Address Queue

In this circuit, the function is enabled for port 2, not for ports 0, 1, 3, 4.

When enabled, there are two separate read address queues in the XPI block—red queue and blue queue.

Both read queues request independently to the PA.

Red queue can be used for one traffic class only and associated to Region 2 in the QoS mapper (HPR or VPR).

Blue queue can be used for two traffic classes and associated to Region 0 and Region 1 in the QoS mapper (VPR and/or LPR).

There are separate time-outs for the blue and red queues. Timers are started to down count when the transaction is accepted in the XPI, which are then forwarded to DDRC together with the command.

17.5.5.1.5 ID Collisions

Due to the dynamic nature of the QoS with respect to the IDs, two transactions of the same ID can potentially exist in two queues. The collisions can be classified as:

- Blue after Red (BAR): a command with ID x is presented to the blue queue where a command with the same ID x is outstanding in the red queue.
- Red after Blue (RAB): a command with ID x is presented to the red queue where a command with same ID x is outstanding in the blue queue.

In the XPI, for both collision types, the ordering consistency within a given ID is preserved by stalling the incoming transaction to be pushed into the address queue it is mapped to until the colliding transaction in the opposite queue exists. This is required for functional correctness.

In case of any ID collision, the UDDRC does not speed up the drain of the stored transactions in the address queue causing collision even if the incoming transaction is HPR. In other words, address queues request arbitration to the Port Arbiter with their normal mapped priorities.

17.5.5.2 VPR/VPW Timeout

There are separate timeouts for the blue and red queues for read transactions, and one timeout for write transactions. Timers are started to down count when the transaction is accepted in the XPI, which are then forwarded to DDRC together with the command.

Timeouts are set per port and queue using PCFGQOS1_n and PCFGWQOS1_n registers (see UDDRC_PCFGQOS1_0 and UDDRC_PCFGWQOS1_0).

For PCFGQOS1_n register:

- RQOS_MAP_TIMEOUTB: specifies the timeout value for transactions mapped to the blue queue.
- RQOS_MAP_TIMEOUTR: specifies the timeout value for transactions mapped to the red queue.

For PCFGWQOS1_n register:

- WQOS_MAP_TIMEOUT: specifies the timeout value for write transactions.

When expired, VPR/VPW transactions from XPI are tagged as expired-VPR or expired-VPW instead of LPR/NPW during normal priority transaction flow. The Port Arbiter treats these transactions with the highest priority (priority0). In addition, the Port Arbiter asserts hif_go2critical_lpr signal to the DDRC if there are no LPR credits available (LPR store of the read CAM is full) when an expired-VPR port is pending, and hif_go2critical_wr signal to the DDRC if there are no write credits available when an expired-VPW is pending. If an expired-VPW is issued as RMW at the HIF interface, the Port Arbiter asserts hif_go2critical_lpr signal to the DDRC if there are no LPR credits hif_go2critical_lpr and hif_go2critical_wr are asserted at the same time, priority in the DDRC is given to the read.

If VPR/VPW timeout registers are set to 0, the VPR/VPW transactions expire immediately as they enter the DDRMC, thereby making them the highest priority transaction class within the device. If multiple VPR/VPW transactions expire at the same time in the XPI, they are executed by the PA in round-robin order.

17.5.5.3 Registers Related to QoS

The following are the registers related to the QoS:

- UDDRC_PCFGQOS0_x
- UDDRC_PCFGQOS1_x
- UDDRC_PCFGWQOS0_x
- UDDRC_PCFGWQOS1_x

Five AXI ports are available, so x can vary from 0 to 4. For more information about these registers, see Register Descriptions.

17.5.6 Bypass Operation

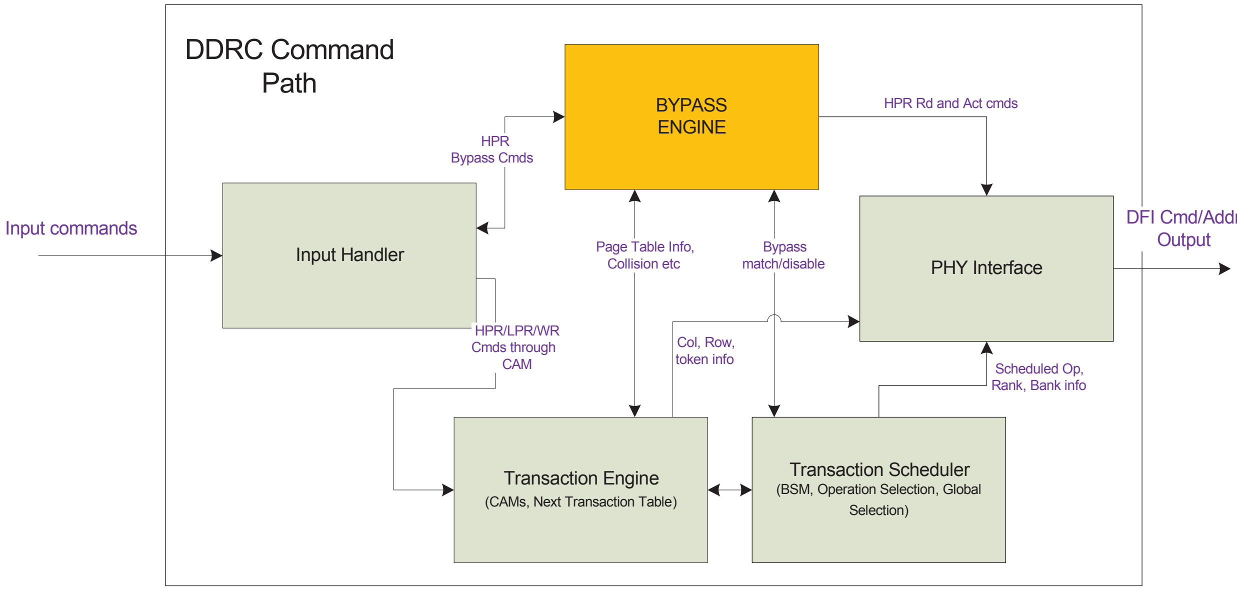
The bypass engine in the UDDRC handles high-priority, low-latency read requests.

The UDDRC supports two priority levels for read commands – high priority reads (HPR) and low priority reads (LPR). The Read CAM can be split into low and high priority sections using the SCHED.lpr_num_entries register. Generally, all the read commands of the controller are stored in the Read CAM. HPR commands are directed to the HP section and LPR commands directed to the LP section. The commands are then scheduled based on several factors such as priority, page match, and oldest entry. The HPR commands in the CAM are given higher priority over the LPR commands by the scheduling engine. However, the HPR commands can be present in the CAM for longer duration due to several factors.

If the bypass operation is enabled, the HPR commands can bypass the CAM and go directly to the PHY interface engine, thereby saving several clocks in latency.

The bypass and the CAM paths are shown in the following figure.

Figure 17-4. Bypass and CAM Paths



There are two types of bypass operations:

- Activate Bypass operation: if a HP Read command comes to a closed page, then the Activate request for this command is sent through the Bypass path. The command is still sent to the CAM and the Read part of this request happens through the CAM scheduling logic.
- Read Bypass operation: if a HP Read command comes to a page that is already open, then the Activate request is not necessary for this command. The Read is sent through the Bypass path.

Even if the input command satisfies the above mentioned conditions for the bypass operation, the command may not go through the Bypass path.

There are several reasons for this:

- The Controller is in Write mode.
- The Controller is performing critical maintenance commands such as Refresh, ZQ Calibration, MRW/MRR, and Critical Pre-charge.
- The Controller is handling a collision scenario.
- The Controller is performing a DFI Control Update request.

The idle latency through the DDRC for an Activate or a Read command going through the CAM path is 4 cycles. The idle latency when the same command goes through the Bypass path is 2 cycles.

17.5.6.1 Enabling Bypass Operation

To enable the bypass operation, DBG0.disable_act_bypass and DBG0.disable_rd_bypass registers should be 0 (see Register Descriptions).

17.5.6.2 Registers Related to Bypass Operation

The following are the registers related to the Bypass Operation:

- UDDRC_DBG0.DISABLE_ACT_BYPASS
- UDDRC_DBG0.DISABLE_RD_BYPASS

For more information about these registers, see Register Descriptions.

17.5.7 Burst Mode Operation

An abnormal read or write transaction on the host interface contains enough data for one DDR BL8 transaction in Full Bus Width mode. The actual burst length used on the DFI interface is controlled by MSTR.burst_rdwr[3:0] and the data bus width (MSTR.data_bus_width). When the PI (PHY interface block before DFI) splits read/write commands, it sends them back-to-back (or as close as is allowed by the protocol). This means that no other reads/writes/activates/

precharges can be sent to that bank, and no refreshes can be sent to that rank, between these commands. However, it is possible for a refresh to be sent to another rank at this time.

17.5.8 Refresh Controls

Refresh can be issued using the auto-refresh feature in the UDDRC or using the direct software request of the refresh command. The RFSHCTL3.dis_auto_refresh register bit selects the refresh method (see Register Descriptions). When this bit is set to 1, UDDRC uses direct software request of refresh command. When it is set to 0, the internal auto-refresh feature is used.

The purpose of refresh control is to:

- Reduce the bandwidth impact of refresh cycles
- Increase the likelihood of refreshes being serviced during idle periods
- Provide fine-gain control of the trading-off the above benefits (to gather refreshes) versus the increased worst case latencies associated with gathering refreshes

17.5.8.1 Refresh Using Direct Software Request of Refresh Command

Follow these steps to put the UDDRC in direct software request of Refresh Command mode:

1. Set the RFSHCTL3.dis_auto_refresh bit to 1. When the register bit is set, the UDDRC checks for any pending refreshes. Any pending refreshes are issued right away using the ‘critical refresh’ feature inside the UDDRC. After these refreshes are issued, all the refresh timers inside the UDDRC are reset to 0. They are re-activated only when the auto-refresh feature is enabled.
2. The SoC core must keep track of the refresh requirements of the SDRAM.
3. The refresh command can be issued by setting the register bits DBGCMD.rank*_refresh to 1 (see Register Descriptions). When the rank*_refresh request is stored in the UDDRC, the corresponding register bit is automatically cleared. The SoC core can initiate a rank*_refresh operation only if DBGSTAT.rank*_refresh_busy is low. The UDDRC issues refresh to the SDRAM at the earliest.
4. Software-driven refresh commands for each rank are loaded into a 9-entry buffer, and are issued by the UDDRC on the DFI as soon as it is legal to do so (the UDDRC controller must wait $t_{RFC(min)}$ between each refresh request). If the buffer saturates, the DBGSTAT.rank*_refresh_busy remains asserted to prevent software from initiating further refreshes.

17.5.8.2 Refresh Using Auto Refresh Feature Inside the UDDRC

The UDDRC provides advanced refresh controls. Besides fully-configurable refresh constraints ($t_{RFC(min)}$ and t_{REFI}), the UDDRC can also be programmed to gather refreshes to each rank of SDRAM to reduce the bandwidth consumed by refreshes and to increase the likelihood that refreshes can be serviced during an idle period.

Fine-grain control of the refreshes ensures these benefits can be balanced against worst case latencies associated with servicing refreshes together. Staggered refresh timers for multi-rank configurations of the UDDRC allow transactions to continue to other ranks while refreshes are taking place to just one rank.

To minimize the worst case impact of a forced refresh cycle, the UDDRC can be programmed to issue single refreshes at a time by forcing RFSHCTL0.refresh_burst = 0 (see Register Descriptions). It can be programmed to burst up to 8 refreshes (RFSHCTL0.refresh_burst = 7). In LPDDR2/3 mode, with per-bank refresh enabled (RFSHCTL0.per_bank_refresh = 1), the maximum refresh burst supported by UDDRC is 64 (RFSHCTL0.refresh_burst = 63). Burst refresh can be used to minimize the bandwidth lost to closing pages for refresh and to increase the likelihood that refreshes can be serviced during idle periods. It can also be programmed to any number in between to trade-off the benefits of each.

17.5.8.2.1 Single Refresh

When using single refresh (RFSHCTL0.refresh_burst = 0), the UDDRC issues refreshes every time the refresh timer (t_{REFI}) expires. This is the optimal mode of operation for systems that minimize the maximum latency associated with refresh cycles.

17.5.8.2.2 Burst Refresh

When burst refresh is enabled (RFSHCTL0.refresh_burst > 0), the UDDRC issues refreshes in bursts of (RFSHCTL0.refresh_burst+1) refreshes at one time. Bursting refreshes reduce the total latency associated with those refreshes by reducing the number of precharges and activates required for refresh, as banks must be precharged only once to perform the entire group of refreshes, instead of once for each refresh.

17.5.8.2.3 Speculative Refresh

When the burst refresh is enabled (RFSHCTL0.refresh_burst > 0), you can also make use of a feature called speculative refresh.

Burst refresh is implemented by counting the number of times t_{REFI} expires and issuing a group of refreshes when that number reaches the refresh burst number. Once t_{REFI} has expired at least once, the UDDRC can also perform speculative refreshes. This is done by automatically inserting refreshes when there are no transactions pending in the CAM to a rank/bank address.

The RFSHCTL0.refresh_to_x1_x32 register determines how long the UDDRC must be idle before considering inserting these speculative refreshes. Each time a speculative refresh is performed, the count of t_{REFI} expirations is decremented, thereby increasing the time before a burst of refreshes is required. This also ensures that speculative refreshes never occur more often than is required to keep the SDRAM properly refreshed.

If a new read or write transaction is accepted by the UDDRC during speculative refresh, the UDDRC services it as soon as legally possible. Most often it entails waiting for the required NOP cycles after a refresh before performing an activate and then servicing the read or write. If the UDDRC has begun closing pages for a speculative refresh but has not yet issued the refresh when the new transaction arrives, the speculative refresh is canceled.

17.5.8.2.4 Per-bank Refresh (LPDDR2/LPDDR3 only)

If RFSHCTL0.per_bank_refresh is set to 1 (see Register Descriptions), the UDDRC performs per-bank refreshes instead of all-bank refreshes. In this case, RFSHTMG.t_rfc_nom_x1_x32 and RFSHTMG.t_rfc_min should be set to the appropriate values for per-bank refresh (tREFIpb and tRFCpb respectively). In this mode, the UDDRC keeps track of which bank is being refreshed at any time, and is able to schedule commands to other banks immediately before and after the per-bank refresh commands, resulting in potential efficiency gains. To improve the accuracy of per-bank refresh timing, set RFSHTMG.t_rfc_nom_x1_sel to 1 and program RFSHTMG.t_rfc_nom_x1_x32 accordingly (see Register Descriptions).

If the refresh burst (RFSHCTL0.refresh_burst) is programmed with a large value in per-bank Refresh mode, and bursting per-bank refreshes start, it may take a longer time than bursting all-bank refreshes. The three possible cases where bursting per-bank refreshes may occur are:

1. Random traffic
   The UDDRC may not be able to issue a speculative refresh because the CAM always has transactions on some banks. This can be avoided with appropriate register settings. For example, setting smaller values to RFSHCTL0.refresh_burst or RFSHCTL0.refresh_to_x1_x32.
2. Toggling RFSHCTL3.refresh_update_level
3. Hardware Fast Frequency Change (HWFFC)

After operation 2 or 3, the UDDRC starts bursting refreshes to compensate for any updated refresh timing parameters.

17.5.8.2.5 Constraints on refresh_timerX_start_value_x32

The refresh_timerX_start_value_x32 register fields control the relative timing of refreshes to different ranks.

The register fields must obey the following constraint:

refresh_timerX_start_value_x32 + RoundUp(RFSHTMG.t_rfc_min/32) < RFSHTMG.t_rfc_nom_x1_x32

If RFSHTMG.t_rfc_nom_x1_sel is set, RoundDown(RFSHTMG.t_rfc_nom_x1_x32/32) should be used in the above constraint.

Also, note that the register field refresh_timerX_start_value_x32 cannot be changed after initialization. So the value of the register RFSHTMG.t_rfc_nom_x1_x32 used above should be anticipated as the minimum value which will be used, considering frequency change, fine granularity refresh, etc.

For LPDDR2/3, if DERATEEN.derate_enable =1, use minimum RFSHTMG.t_rfc_nom_x1_x32/4 as value in calculation of refresh_timerX_star_value_x32 to consider possible effect of derating in high temperature case.

17.5.8.2.6 Registers Related to Refresh Controls

- UDDRC_RFSHCTL0
- UDDRC_RFSHCTL1
- UDDRC_RFSHCTL2
- UDDRC_RFSHCTL4

- UDDRC_RFSHCTL3
- UDDRC_RFSHTMG
- UDDRC_RFSHTMG1

For more information about these registers, see Register Descriptions.

17.5.9 ZQ Calibration

This feature is applicable to DDR3 and LPDDR2/LPDDR3. The UDDRC controller uses the ZQ calibration command to calibrate SDRAM RON (Resistor On) and ODT (On-Die Termination) values over PVT (Process, Voltage, Temperature). DDR3 and LPDDR2/LPDDR3 SDRAMs need more time to calibrate RON and ODT at initialization and relatively less time to perform periodic calibrations. For more information, refer to the following specifications:

- DDR3 (JEDEC JESD79-3F)
- LPDDR2 (JEDEC JESD209-2E)
- LPDDR3 (JEDEC JESD209-3B)

17.5.9.1 DDR3 Devices

ZQCL (ZQ Calibration Long) command is used as follows:

- To perform the initial calibration during the power-up initialization sequence. This command is allowed a timing period of tZQinit determined by INIT5. dev_zqinit_x32 (see Register Descriptions) to perform full calibration and the transfer of values.
- To perform long ZQ Calibration after exiting from Self-refresh mode. This command is allowed a timing period of tZQOPER, determined by ZQCTL0._t_zq_long_nop. This command is automatically issued when ZQCTL0.dis_srx_zqcl (ZQCTL0.dis_mpsmx_zqcl) is set to 0. To disable issuing ZQCL after self-refresh exit, set ZQCTL0.dis_srx_zqcl (ZQCTL0.dis_mpsmx_zqcl) to 1.

ZQCS (ZQ Calibration Short) command is used to perform periodic calibration to account for VT (Voltage/ Temperature) variations. A shorter timing window is provided to perform calibration and transfer of values as defined by the timing parameter tZQCS (determined by ZQCTL0._t_zq_short_nop). ZQCS can be performed automatically on a regular interval or through direct software request. For more information, see Automatic and Software Initiated ZQCS.

In DDR3 mode, the ZQ Calibration commands are sent out encoded on the DFI command bus as mentioned in the JEDEC Specification. The UDDRC performs no other activities for the duration of tZQinit, tZQOPER and tZQCS. The quiet time on the SDRAM channel helps in accurate calibration of SDRAM RON and ODT. All banks are precharged and t_{RP} met before the UDDRC issues the ZQ Calibration commands.

17.5.9.2 LPDDR2/LPDDR3 Devices

The UDDRC supports all the ZQ Calibration commands that are supported by the JEDEC Specification.

- ZQInit (ZQ Initial Calibration) command performs the initial calibration during the power-up initialization sequence. This command is allowed a time period of tZQinit (determined by INIT5.dev_zqinit_x32).
- ZQCL (ZQ Long Calibration) command performs long ZQ Calibration after exiting from Self-refresh mode. This command is allowed a timing period of tZQCL (determined by ZQCTL0._t_zq_long_nop). This command is issued automatically when ZQCTL0. dis_srx_zqcl is set to 0. To disable issuing of ZQCL after self-refresh exit, set ZQCTL0. dis_srx_zqcl to 1.
- ZQCS (ZQ Short Calibration) command is used to perform periodic calibration to account for VT (Voltage/ Temperature) variations. A shorter timing window is provided to perform calibration and transfer of values as defined by the timing parameter tZQCS (determined by ZQCTL0._t_zq_short_nop). ZQCS can be performed automatically on a regular interval or through direct software request. For more information, see Automatic and Software Initiated ZQCS.
- ZQReset (ZQ Calibration Reset) command is used to reset the RON calibration to a default accuracy of ±30% across process, voltage and temperature. This command is used to ensure RON accuracy of ±30% when ZQCS and ZQCL are not used and is allowed a time period of tZQRESET, determined by ZQCTL1.t_zq_reset_nop. The command is issued by using the registers ZQCTL2.zq_reset and ZQSTAT.zq_reset_busy (see Register Descriptions). For more information, see LPDDR2/LPDDR3 ZQ Reset Command.

In LPDDR2/LPDDR3 mode, ZQ Calibration commands are sent out as Mode Register Write commands to the DRAM. The MRW is done to MR10 and the calibration codes for the different commands are as follows:

ZQInit – 0xFF, ZQCL – 0xAB, ZQCS – 0x56 and ZQRest – 0xC3

The UDDRC performs no other activities for the duration of tZQinit, tZQCL, tZQCS, and tZQRESET. The quiet time on the SDRAM channel helps in accurate calibration of SDRAM RON and ODT. All banks are precharged and t_{RP} met before the UDDRC issues the ZQ Calibration commands.

17.5.9.3 Automatic and Software Initiated ZQCS

The UDDRC issues a ZQCS command in the following ways:

- Automatic ZQCS by the UDDRC: in this case, UDDRC sends ZQCS commands to SDRAM periodically. The interval is determined by ZQCTL1.t_zq_short_interval_x1024. This method is used if ZQCTL0.dis_auto_zq is set to 0 (see Register Descriptions).
- ZQCS using direct software request: in this case, the SoC core sends a ZQCS command through software by setting DBGCMD.zq_calib_short to 1. When the ZQCS request is stored in the UDDRC, the register bit is automatically cleared. It is recommended not to set DBGCMD.zq_calib_short signal, in Init, Self-refresh, Deep Power-down operating modes or Maximum Power Saving mode (MPSM). The SoC core can initiate a ZQCS operation only if DBGSTAT.zq_calib_short_busy is low. The DBGSTAT.zq_calib_short_busy signal goes high in the clock after the UDDRC accepts a ZQCS request. It goes low when the ZQCS operation is initiated in the UDDRC. For proper SDRAM operation, user/SoC core should schedule this command frequently. This method is used if ZQCTL0.dis_auto_zq is set to 1.

For self-refresh, command is scheduled after SR/SR-Powerdown is exited. For Deep Power Down or MPSM, command is not scheduled although ZQSTAT.zq_calib_short_busy is de-asserted.

17.5.9.4 LPDDR2/LPDDR3 ZQ Reset Command

In LPDDR2/LPDDR3 mode, the ZQ Reset command is issued by setting ZQCTL2.zq_reset to 1 (see Register Descriptions). When the ZQ Reset operation is complete, the UDDRC automatically clears this register bit. In LPDDR2/LPDDR3 mode, it is recommended not to set this signal in Init, Self-refresh or Deep Power-down operating modes. The SoC core can initiate a ZQ Reset operation only if ZQSTAT.zq_reset_busy is low. This signal goes high in the clock after the UDDRC accepts a ZQ Reset request. It goes low when the ZQ reset command is issued to the SDRAM and the associated NOP period is completed.

For self-refresh, command is scheduled after SR/SR-Powerdown is exited. For Deep Power Down, command is not scheduled although ZQSTAT.zq_reset_busy is de-asserted.

17.5.9.5 Registers Related to ZQ Calibration

- UDDRC_ZQCTL0
- UDDRC_ZQCTL1
- UDDRC_ZQCTL2
- UDDRC_ZQSTAT

For more information about these registers, see Register Descriptions.

17.5.10 ODT Control

By default, ODT to memories is driven to all zeros.

The register inputs control the following:

- ODTMAP.rank*_wr_odt: the value desired for ODT following a write command (see Register Descriptions).
- ODTCFG.wr_odt_delay: the number of cycles to delay following a write command before driving the programmed values for write ODT, which depends primarily on CAS latency.
- ODTCFG.wr_odt_hold: the number of cycles to hold the programmed write value after it is first driven.
- ODTMAP.rank*_rd_odt: the value desired for ODT following a read command.
- ODTCFG.rd_odt_delay: the number of cycles to delay following a read command before driving the programmed values for write ODT, which depends primarily on CAS latency.
- ODTCFG.rd_odt_hold: the number of cycles to hold the programmed read value after it is first driven.

Notes:
1. ODT control for memories is not required for LPDDR2. It is a DDR2/DDR3/LPDDR3 specific feature.
2. For LPDDR2, set ODTMAP.rank*_wr/rd_odt to all zeroes.

All of these must be set by the SoC core before taking the UDDRC out of reset. They are then applied to every read or write issued by the UDDRC.

For recommended settings for ODT-related registers in each protocol, see Register Descriptions.

17.5.10.1 Registers Related to ODT Control

The following are the registers related to the ODT Control:

- UDDRC_ODTCFG
- UDDRC_ODTMAP

For more information about these registers, see Register Descriptions.

17.5.11 High-Level SDRAM Initialization Procedure

Proceed as follows to initialize the SDRAM. Code example is given in provided software (Linux, Harmony).

Note: This sequence can also be used as a re-initialization phase from Self-refresh mode. In such case, some commands must be skipped or added. See SDRAM Self-Refresh Mode in the section "Electrical Characteristics" for more details.

17.5.11.1 Global Configuration

1. Update the SFRBU DDR Power Control register (SFRBU_DDRPWR) to take the DDR I/Os out of Retention state.
   – Wait for the command to complete by polling bit SFRBU_DDRPWR.RETENTION until it sets to 1.
2. Update the RSTC Generic Reset register (RSTC_GRSTR) to reset the DDR PHY and DDR controllers.
3. Perform a dummy read to allow 128 cycles for synchronization.
4. Update RSTC_GRSTR to release the DDR3PHY reset.
5. Perform a dummy read to allow 128 cycles for synchronization.

17.5.11.2 Step 1

1. Clear the DFI Miscellaneous Control register (UDDRC_DFIMISC).
2. Update the Host register 0 (UDDRC_MSTR) to enter details about the DDR used, indicate whether BURSTCHOP is used and set the required burst length.
3. If using LPDDR2/3 above 85°C: program the Temperature Derate Enable (UDDRC_DERATEEN) and Temperature Derate Interval (UDDRC_DERATEINT) registers.
4. Update the Low Power Control register (UDDRC_PWRCTL) to enable assertion of en_dfi_dram_clk_disable.
5. Update the Low Power Timing register (UDDRC_PWRTMG) with:
   – Number of clocks with command channel idle before UDDRC automatically puts SDRAM into Self-refresh mode
   – Number of clocks with command channel idle before UDDRC automatically puts SDRAM into Power-down mode
   – Minimum number of DFI clocks in Deep Power-down mode once PWRCTL.DEEPPOWERDOWN_EN is de-asserted
6. Update the Hardware Low Power Control register (UDDRC_HWLPCTL) to:
   – enable exiting from Automatic Clock Stop, Automatic Power-down or Automatic Self-refresh mode,
   – program the number of DFI clock cycles necessary once the command channel is idle before the active_ddrc output is driven low.
7. Update the Refresh Control register 0 (UDDRC_RFSHCTL0) with:
   – Refresh burst value
   – Refresh period
   – Critical refresh threshold margin value
8. Clear the UDDRC Refresh Control register 3 (UDDRC_RFSHCTL3).
9. According to the memory type, fill the following fields in the UDDRC Refresh Timing register (UDDRC_RFSHTMG):
   – T_RFC_MIN
   – T_RFC_NOM_X1_X32
10. Clear the CRC Parity Control register 0 (UDDRC_CRCPARCTL0).
11. Update SDRAM Initialization registers 0 to 5 (UDDRC_INIT0 to UDDRC_INIT5).

12. Clear the DIMM Control register (UDDRC_DIMMCTL).
13. Update SDRAM Timing registers 0 to 8 (UDDRC_DRAMTMG0 to UDDRC_DRAMTMG8) (and UDDRC_DRAMTMG14 for LPDDR1/2/3).
14. Update ZQ Control registers 0 and 1 (UDDRC_ZQCTL0 and UDDRC_ZQCTL1).
15. Update DFI Timing registers 0 and 1 (UDDRC_DFITMG0 and UDDRC_DFITMG1).
16. Update the DFI Low Power Configuration register 0 (UDDRC_DFILPCFG0).
17. Update DFI Update registers 0 to 2 (UDDRC_DFIUPD0 to UDDRC_DFIUPD2).
18. Disable the PHY host interface by clearing the DFI PHY Host register (UDDRC_DFIPHYMSTR).
19. Update the SAR Base Address register 0 (UDDRC_SARBASE0).
20. Update the SAR Size register 0 (UDDRC_SARSIZE0).
21. Update Address Map registers 1 to 11 (UDDRC_ADDRMAP1 to UDDRC_ADDRMAP11) depending on the type of DDR memory used.
22. If a DDR2 or DDR3 is being configured, update the ODT Configuration (UDDRC_ODTCFG) and ODT/Rank Map (UDDRC_ODTMAP) registers.
23. Update Scheduler Control registers 0 and 1 (UDDRC_SCHED and UDDRC_SCHED1).
24. Update the High Priority Read CAM register 1 (UDDRC_PERFHPR1).
25. Update the Low Priority Read CAM register 1 (UDDRC_PERFLPR1).
26. Update the Write CAM register 1 (UDDRC_PERFWR1) so that a low priority write timeout will be the same as a high priority write.
27. Update the Debug register 0 (UDDRC_DBG0).
28. Update the Port Common Configuration register (UDDRC_PCCFG).
29. For each port (0 to 4), update:
    – Port Configuration Read register
    – Port Configuration Write register
    – Port Read QoS Configuration registers 0 and 1
    – Port Write QoS Configuration registers 0 and 1

In case of re-initialization only:

1. Update the SDRAM Initialization register 0 (UDDRC_INIT0) to skip the SDRAM initialization while keeping the controller in Self-refresh mode.
2. Update UDDRC_PWRCTL to keep the memory in Self-refresh mode.
3. Clear UDDRC_DFIMISC.

17.5.11.3 Step 2

Update the RSTC Generic Reset register (RSTC_GRSTR) to bring the DDR Controller out of Reset state.

17.5.11.4 Step 3

Update:

1. DDR3PHY DRAM Configuration register
2. DDR3PHY PHY General Configuration register
3. DDR3PHY PHY Timing registers 0 to 2
4. DDR3PHY Mode registers 0 to 3
5. DDR3PHY ODT Configuration register
6. DDR3PHY DRAM Timing Parameters registers 0 to 2
7. DDR3PHY DDR System General Configuration register
8. DDR3PHY DATX8 Common Configuration register
9. DDR3PHY ZQ Impedance Control register 1 (if a DDR2 or DDR3 is being configured)

In case of re-initialization only: bypass the impedance calibration of ZQ control blocks by setting ZCALBYP in the DDR3PHY Initialization register (DDR3PHY_PIR).

17.5.11.5 Step 4

1. Wait for the DDR3PHY initialization to complete by polling the IDONE bit in the DDR3PHY PHY General Status register (DDR3PHY_PGSR).

2. In case of re-initialization only: override the ZQ Control impedance calibration by:
   – writing 1 to ZDEN in DDR3PHY ZQ Impedance Control register 0 (DDR3PHY_ZQ0CR0),
   – filling the ZQ impedance field with the previously saved impedance values,
   – restoring data corrupted by the training.

17.5.11.6 Step 5

1. Start the DDR system initialization. Use the controller by setting the DDR3PHY_PIR.INIT and DDR3PHY_PIR.CTLDINIT bits.
2. Wait for the DDR PHY initialization to complete by polling DDR3PHY_PGSR.IDONE.
3. In case of re-initialization only:
   – Update the SFRBU DDR Power Control register (SFRBU_DDRPWR) to clear the SFRBU_DDRPWR.RETENTION bit and take the DDR I/Os out of Retention state.
   – Wait for the command to complete by polling the SFRBU_DDRPWR.RETENTION bit until it sets to 1.
   – Remove the ZQ calibration override by writing 0 to DDR3PHY_ZQ0CR0.ZDEN.
   – Trigger ZQ calibration.
   – Wait for calibration to finish.
   – Read DDR3PHY_ZQ0CR0.ZCAL to check for a ZQ calibration error.
4. For full strength (18-Ohm) DDR2, perform a custom calibration. Refer to the provided sample initialization code.

17.5.11.7 Step 6

Update the Software Register Programming Control Enable register (UDDRC_SWCTL) to enable quasi-dynamic register programming.

17.5.11.8 Step 7

Update UDDRC_DFIMISC to set the DFI_INIT_COMPLETE_EN bit.

17.5.11.9 Step 8

Set UDDRC_SWCTL.SW_DONE to indicate that UDDRC programming is completed.

17.5.11.10 Step 9

In the Software Register Programming Control Status register (UDDRC_SWSTAT), poll the SW_DONE_ACK bit until it changes to 1, acknowledging that programming is complete.

In case of re-initialization only:

1. Update UDDRC_PWRCTL.SELFREF_SW to exit Self-refresh mode.
   – Wait until OPERATING MODE, in the Operating Mode Status register (UDDRC_STAT), changes to 0x1.

17.5.11.11 Step 10

Wait until UDDRC_STAT.OPERATING MODE changes to 0x1.

17.5.11.12 Step 11

1. Update UDDRC_RFSHCTL3 to disable auto-refreshes by setting DIS_AUTO_REFRESH to 1.
2. Toggle UDDRC_RFSHCTL3.REFRESH_UPDATE_LEVEL.
3. Update UDDRC_SWCTL to enable quasi-dynamic register programming.
4. Update UDDRC_DFIMISC to set DFI_INIT_COMPLETE_EN.
5. Set UDDRC_SWCTL.SW_DONE to indicate that UDDRC programming is completed.
   – Poll the UDDRC_SWSTAT.SW_DONE_ACK bit until it changes to 1, acknowledging that programming is complete.
6. Update Port Control registers for ports 0 to 4 (UDDRC_PCTRL_0 to UDDRC_PCTRL_4) to enable the AXI ports.

17.5.11.13 Step 12

In the DDR3PHY PHY General Configuration register (DDR3PHY_PGCR), set the RANKEN field to enable training for rank 1.

17.5.11.14 Step 13

Update DDR3PHY_PIR to launch RV and QS trainings.

- Wait for the DDR3PHY initialization to complete by polling DDR3PHY_PGSR.IDONE.

17.5.11.15 Step 14

Read DDR3PHY_PGSR to check whether the training was successful or if errors occurred.

In case of re-initialization only: restore data corrupted by the training using the DCU command cache.

17.5.11.16 Step 15

1. Update UDDRC_RFSHCTL3 to enable auto-refreshes by clearing DIS_AUTO_REFRESH.
2. Toggle UDDRC_RFSHCTL3.REFRESH_UPDATE_LEVEL.
3. Update UDDRC_SWCTL to enable quasi-dynamic register programming.
4. Update UDDRC_DFIMISC to set DFI_INIT_COMPLETE_EN.
5. Set UDDRC_SWCTL.SW_DONE to indicate that UDDRC programming is completed.
   – Poll the UDDRC_SWSTAT.SW_DONE_ACK bit until it changes to 1, acknowledging that programming is complete.
6. Update Port Control registers for ports 0 to 4 (UDDRC_PCTRL_0 to UDDRC_PCTRL_4) to enable the AXI ports.

17.5.11.17 Step 16

Program the TrustZone controllers to enable SDRAM use.

This initialization allows TZC400 to accept all accesses (secure or not secure) in the DDR address space.

Note: If TZC400 is not programmed, all accesses are blocked (without any response error).

17.5.12 Mode Register Reads and Writes

This section explains how to perform mode register reads and writes through software. Mode Register Reads (MRR) are applicable only to LPDDR2/LPDDR3, and are used to read configuration and status data from mode registers in the SDRAM. Mode Register Writes (MRW or MRS) are applicable to all supported DDR protocols, and are used to write configuration data to mode registers in the SDRAM. Access to the mode register is initiated by programming the MRCTRL0 and MRCTRL1 registers (see Register Descriptions). This must be done in three steps:

1. Poll MRSTAT.mr_wr_busy until it is ‘0’. This checks that there is no outstanding MR transaction. No writes should be performed to MRCTRL0 and MRCTRL1 if MRSTAT.mr_wr_busy = 1.
2. Write the MRCTRL0.mr_type, MRCTRL0.mr_addr, MRCTRL0.mr_rank and (for MRWs) MRCTRL1.mr_data to define the MR transaction.
3. In a separate APB transaction, write the MRCTRL0.mr_wr to 1. This bit is self-clearing, and triggers the MR transaction. The UDDRC then asserts the MRSTAT.mr_wr_busy while it performs the MR transaction to SDRAM, and no further accesses can be initiated until it is deasserted.

17.5.12.1 Mode Register Write

MRW transactions can be specified for either any single rank, or any combination of several ranks, by programming MRCTRL0.mr_rank.

If any part of the SDRAM’s MR register is updated by software, it is also the responsibility of the software to update corresponding INIT*.*mr* so that it is aligned to the SDRAM’s MR register. For LPDDR2/3, INIT* registers are used to perform SDRAM initialization after DPDX.

17.5.12.2 Mode Register Read

MRR transactions should only be performed to one rank at a time, to avoid bus contention. When an MRR is performed, the Mode register contents are available on hif_mrr_data, qualified by hif_mrr_data_valid, after the Mode register read command is issued by the UDDRC to the SDRAM. Note that the entire width of the SDRAM data is mapped to hif_mrr_data. You must select the appropriate byte, depending on whether the bytes are swapped on the board or not.

17.5.12.3 Registers Related to Mode Register Reads and Writes

The following are the registers related to the Mode register reads and writes:

- UDDRC_MRCTRL0
- UDDRC_MRCTRL1
- UDDRC_INIT0
- UDDRC_INIT1
- UDDRC_INIT2
- UDDRC_INIT3
- UDDRC_INIT4
- UDDRC_INIT5
- UDDRC_INIT6
- UDDRC_INIT7

For more information about these registers, see Register Descriptions.

17.5.13 2T Memory Command Timing

Memory command duration is one memory clock cycle, which is referred to as 1T command timing.

Optionally, to provide sufficient address setup time in heavily loaded memory bus configurations, memory address and memory command are asserted for two memory clock cycles which is referred to as 2T command timing.

Figure 17-5. 1T Memory Command Timing

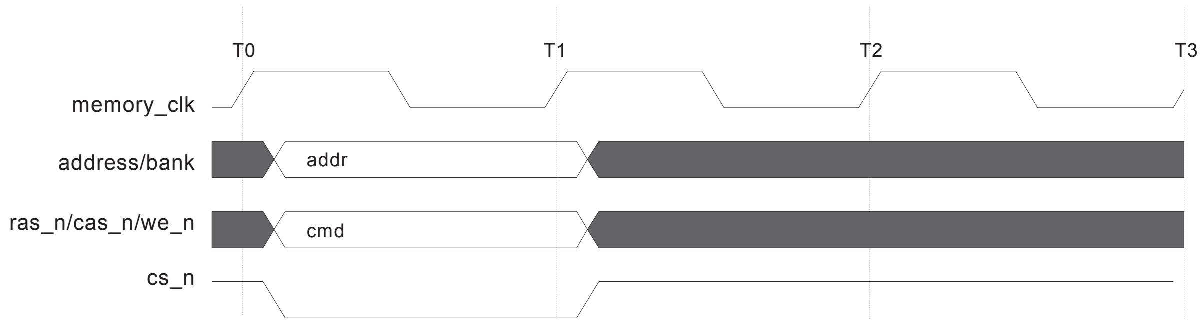
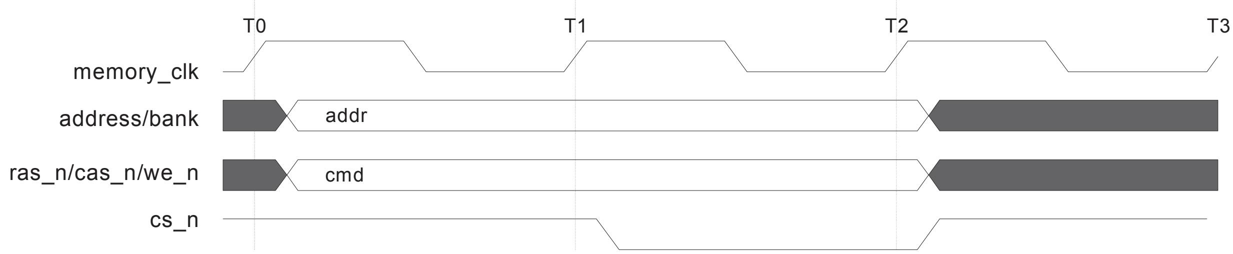


Figure 17-6. 2T Memory Command Timing



Notes:
1. 2T mode is not supported when BL2 is selected.
2. 2T mode is not supported in LPDDR2/LPDDR3 modes.

Enabling 2T Memory Command Timing:

This feature is enabled using the register MSTR.en_2t_timing_mode.

For more information about these registers, see Register Descriptions.

17.5.14 Power Saving Features

The UDDRC supports various SDRAM power saving modes, such as Precharge Power-down, Self-refresh, Deep Power-down, Maximum Power Saving, and support for disabling clock to the DRAM through dfi_dram_clk_disable.

Refer to Self-refresh mode entering and exiting procedures in the section "Electrical Characteristics".

17.5.14.1 SDRAM Power Saving Features

17.5.14.1.1 Precharge Power-down

Entering Precharge Power-down

When PWRCTL.powerdown_en = 1 (see Register Descriptions), UDDRC automatically enters precharge power-down when the period specified by PWRTMG.powerdown_to_x32 has passed while the UDDRC is idle (except for issuing refreshes).

Entering Precharge Power-down mode involves the following steps:

1. If there is a self-refresh exit previously, wait for at least one refresh command (or 8 per-bank refresh commands if LPDDR2/3 per-bank refresh is enabled) to all active ranks. Auto-refresh logic must be enabled, or refresh should be issued using direct software requests of refresh command through DBGCMD.rank*_refresh.
2. Precharging (closing) all open pages. Pages are closed one-at-a-time in no specified order.
3. Waiting for t_{RP} (row precharge) idle period.
4. Issuing the command to enter precharge power-down (NOP/deselect with CKE = 0).
5. This step occurs only if DFI low-power interface for power-down is enabled (DFILPCFG0.dfi_lp_en_pd). Attempts an entry to Low-power mode through DFI low-power interface and with dfi_lp_wakeup set by DFILPCFG0.dfi_lp_wakeup_pd. The low power entry attempt is delayed with DFITMG0.dfi_t_ctrl_delay + DRAMTMG7.t_ckpde clock cycles. This is needed to satisfy SDRAM timings related to disabling clocks when the PHY is programmed to gate the clock to save maximum power.

If the UDDRC receives a read or write request from the SoC core during step 2 or step 3 above, the power-down entry is immediately aborted. The same is true if PWRCTL.powerdown_en is driven to ‘0’ during step 2 or step 3. Once the power-down entry command is issued, then proper power-down exit is required, as described in “Entering Precharge Power-down” below.

Exiting Precharge Power-down

Once the UDDRC has put the DDR SDRAM device(s) in Precharge Power-down mode, the UDDRC automatically performs the precharge power-down exit sequence for any of the following reasons:

- A refresh cycle is required to any rank in the system.
- The UDDRC receives a new request from the SoC core.
- A self-refresh entry is requested.
- PWRCTL.powerdown_en is set to 0 (see Register Descriptions).

The UDDRC follows these steps when exiting Precharge Power-down mode:

1. Inserting any NOP/deselect commands required to satisfy the t_{CKE} requirement after entering precharge power-down.
2. This step occurs only if DFI Low-power mode entry during power-down entry is successful. Performs an exit from DFI Low-power mode. DFI Low-power mode is exited after the wakeup time specified by DFILPCFG0.dfi_lp_wakeup_pd, but not earlier than DFITMG1.dfi_t_dram_clk_enable + DRAMTMG6.t_ckpdx clock cycles.
3. Issuing the power-down exit command (NOP/deselect with CKE = 1).
4. Issuing NOP/deselect for the period defined by t_{XP}.

Notes:

1. DDR2: Fast Exit versus Slow Exit Active Power-down
The DDR2 Specification describes two different variations on active power-down exit, depending on the programmed value of MR bit 12. As the UDDRC uses precharge power-down rather than active power-down, this programming has no effect on the UDDRC or th SDRAM devices.
2. DDR3: Fast Exit versus Slow Exit Precharge Power-down
The DDR3 Specification describes two different variations on precharge power-down exit, depending on the programmed value of MR0 bit 12. If slow precharge power-down is used (MR0[12] = 0), DRAMTMG1.t_xp should be set to tXPDLL. If fast precharge power-down is used (MR0[12] = 1), DRAMTMG1.t_xp should be set to tXP.
3. Other supported DDR protocols do not specify the difference between fast and slow power-down exit.

17.5.14.1.2 Deep Power-Down

This power saving mode is applicable for LPDDR2 and LPDDR3 devices only.

Entering Deep Power-down

By setting the PWRCTL.deeppowerdown_en bit (see Register Descriptions), you can put the SDRAM device(s) into Deep Power-down mode, if all the following conditions are true:

- The period specified by PWRTMG.powerdown_to_x32 has passed while the UDDRC is idle (except for issuing refreshes).
- PWRCTL.selfref_sw = 0
- PWRCTL.selfref_en = 0
- If HWLPCTL.hw_lp_en = 1, DPD is entered only when the hardware low-power interface has completed a self-refresh exit. (This can be checked by observing STAT.operating_mode and STAT.selfref_type).
- If HWLPCTL.hw_lp_exit_idle_en = 1, DPD is entered only when all bits of cactive_in_ddrc = 0

Entering Deep power-down involves the following steps:

1. If there is a self-refresh exit previously, wait for at least one refresh command (or 8 per-bank refresh commands if LPDDR2/3 per-bank refresh is enabled) to all active ranks. Auto-refresh logic must be enabled, or refresh should be issued using direct software requests of refresh command through DBGCMD.rank*_refresh.
2. Precharging (closing) all open pages. Pages are closed one-at-a-time in no specified order.
3. Waiting for t_{RP} (row precharge) idle period.
4. Issuing the command to enter deep power-down. For multi-rank systems, all chip-selects are asserted so that all ranks enter deep power-down simultaneously. The deep power-down entry command is:
   CKE = 0, CSN = 0, CA0 = 1, CA1 = 1, CA2 = 0
5. This step occurs only if DFI low-power interface for deep power-down is enabled (DFILPCFG0.dfi_lp_en_dpd). It attempts an entry to Low-power mode through DFI low-power interface with dfi_lp_wakeup set by DFILPCFG0.dfi_lp_wakeup_dpd. The low power entry attempt is delayed with DFITMG0.dfi_t_ctrl_delay + DRAMTMG6.t_ckdpde clock cycles, this is needed to satisfy SDRAM timings related to disabling clocks when the PHY is programmed to gate the clock, to save maximum power.

If the UDDRC receives a read or write request from the SoC core during step 1 or step 3 above, the deep power-down entry is immediately aborted. The same is true if PWRCTL.deep_powerdown_en is driven to ‘0’ during step 1 or step 3. Once the deep power-down entry command is issued, proper deep power-down exit is required, as described in the following section.

Note: Contents of SDRAM may be lost upon entry into Deep Power-down mode.

Exiting Deep Power-down

Once the UDDRC puts the DDR SDRAM device(s) in Deep Power-down mode, the UDDRC automatically exits deep power-down and re-runs the initialization sequence when PWRCTL.deeppowerdown_en is reset to 0 (see Register Descriptions). An exit from DFI Low-power mode is performed prior to exiting deep power-down (this occurs only if DFI Low- power mode entry during deep power-down entry is successful). DFI Low-power mode is exited after the wakeup time specified by DFILPCFG0.dfi_lp_wakeup_dpd, but not earlier than DFITMG1.dfi_t_dram_clk_enable + DRAMTMG6.t_ckdpdx clock cycles.

Exiting Deep power-down involves the following steps when SDRAM initialization is performed by the PHY (INIT0.skip_dram_init = 2’b01 or 2’b11):

1. To prevent the UDDRC asserting dfi_cke before the SDRAM initialization is complete, it is necessary to set INIT0.skip_dram_init = 2’b11 before clearing PWRCTL.deeppowerdown_en.
2. If step 1 is performed, to ensure that controller updates will not occur when INIT0.skip_dram_init will be changed back to 2’b01 (which could make DFI bus active when dfi_ctrlupd_req), it is necessary to set DFIUPD0.dis_auto_ctrlupd and DBG1.dis_hif and to stop sending software controller updates before clearing PWRCTL.deeppowerdown_en.
3. Clear DFIMISC.dfi_init_complete_en=“0” register, before clearing PWRCTL.deeppowerdown_en to ensure that the UDDRC will wait until the PHY completes its initialization.
4. Reset PWRCTL.deeppowerdown_en to 0 and poll STAT.operating mode to detect when the UDDRC exits from DPD and then start the SDRAM initialization by setting the PUB_PIR register.

5. Once PHY Init is started and PIR is programmed, set back the old value of skip_dram_init, if it was updated as described in step 1.
6. Poll the relevant PUB's PGSR register to detect when the PUB Initialization is complete.
7. Change back the DFIUPD0.dis_auto_ctrlupd and DBG1.dis_hif values and/or restart sending software controller updates, if they were disabled as described in step 2.
8. Set DFIMISC.dfi_init_complete_en to 1 in order to allow the UDDRC's state machine to exit the Initialization state.

For more information, see SDRAM Initialization Sequence.

17.5.14.1.3 Assertion of dfi_dram_clk_disable

Assertion of dfi_dram_clk_disable occurs only if PWRTL.en_dfi_dram_clk_disable = 1.

dfi_dram_clk_disable is also dependent on the operating mode:

- In DDR2/DDR3, dfi_dram_clk_disable can be asserted only in Self-refresh mode.
- In LPDDR2/LPDDR3, dfi_dram_clk_disable can be asserted in the following modes:
  – in Self-refresh
  – in Power-down
  – in Deep Power-down
  – in Normal mode (“Clock Stop” feature)

The timing of the assertion and de-assertion of dfi_dram_clk_disable in various modes is as follows:

- In Self-refresh or Self-refresh Power-down mode:
  – Asserted at least DFITMG0.dfi_t_ctrl_delay + DRAMTMG5.t_cksre DFITMG1.dfi_t_dram_clk_disable cycles after SRE command.
  – De-asserted at least DFITMG1.dfi_t_dram_clk_enable + DRAMTMG5.t_cksrx - DFITMG0.dfi_t_ctrl_delay cycles before SRX command.
- In Power-down:
  – Asserted at least DFITMG0.dfi_t_ctrl_delay + DRAMTMG7.t_ckpde -DFITMG1.dfi_t_dram_clk_disable cycles after PDE command.
  – De-asserted at least DFITMG1.dfi_t_dram_clk_enable + DRAMTMG7.t_ckpdx -DFITMG0.dfi_t_ctrl_delay cycles before PDX command.
- In Deep Power-down:
  – Asserted at least DFITMG0.dfi_t_ctrl_delay + DRAMTMG6.t_ckdpde - DFITMG1.dfi_t_dram_clk_disable cycles after DPDE command.
  – De-asserted at least DFITMG1.dfi_t_dram_clk_enable + DRAMTMG6.t_ckdpdx -DFITMG0.dfi_t_ctrl_delay cycles before DPDX command.
- In Normal mode (Clock Stop):
  – Asserted at least DFITMG0.dfi_t_ctrl_delay - DFITMG0.dfi_t_dram_clk_disable cycles after any command other than SRE/PDE/DPDE.
  – De-asserted at least DFITMG1.dfi_t_dram_clk_enable + DRAMTMG6.t_ckcsx - DFITMG0.dfi_t_ctrl_delay cycles before any command other than SRX/PDX/DPDX.

For more information about the above modes, see Register Descriptions.

17.5.14.1.4 DLL-off Mode (DDR3)

DLL-off mode enables DDR3 SDRAMs to be operated at low frequencies. The UDDRC supports DLL-off mode, and transitions between DLL-on and DLL-off mode, under software control.

To enable DLL-off mode from initialization, the following register fields must be set:

- INIT3.emr[0], so that the SDRAM mode register is set for DLL-off mode
- MSTR.dll_off_mode = 1
- The DDR PHY should be put in PLL-bypass mode

To perform a transition between DLL-off and DLL-on modes, the software must implement the sequence specified in the JEDEC specification.

Note that geardown and programmable preamble are not supported in DLL-off mode.

17.5.14.2 Power Saving in PHY through DFI Low-Power Interface

Based on whether SDRAM is in Precharge Power-down, Self-refresh, Self-refresh Power-down or Deep Power-down mode (see SDRAM Power Saving Features), the PHY can be placed in power saving modes through the DFI low-power interface.

17.5.14.3 Software Sequence for Removal of Clocks

Software can be used to keep the SDRAM in self-refresh. The AXI and DDRC clocks can be removed when in self-refresh by following the sequence described in the following table.

Note: Dynamic and quasi dynamic registers cannot be programmed if any of the clocks has been removed. Clocks must be turned back on before starting the programming sequence. Also, the clock gating logic must ensure there are no glitches on the clocks when they are removed/enabled.

Table 17-4. Software Clock Removal Sequence

Step	Description	Comment
1	Write 0 to PCTRL_n.port_en	Blocks AXI ports from taking any more transactions
2	Poll PSTAT.rd_port_busy_n = 0 Poll PSTAT.wr_port_busy_n = 0	Waits until all AXI ports are idle
3	Write 1 to PWRCTL.selfref_sw	Causes the system to move to Self-refresh state
4	Poll STAT.selfref_type= 2’b10	Waits until Self-refresh state is entered
5	Remove AXI clocks	–
6	Remove DDRC core clock	–

The clocks should be re-enabled by following the sequence described in the following table.

Table 17-5. Re-enabling the Clocks

Step	Description	Comment
1	Enable AXI clocks	–
2	Enable DDRC core clock	–
3	Write 0 to PWRCTL.selfref_sw	Causes the system to exit from Self-refresh state
4	Poll STAT.selfref_type = 2’b00	Waits until Self-refresh state is exited
5	Write 1 to PCTRL_n.port_en	AXI ports are no longer blocked from taking transactions

17.5.14.4 Power Removal Flow

Table 17-6. Power Removal

Step	Description	Comment
1	Write 0 to PCTRL_n.port_en	Blocks AXI ports from taking any more transactions
2	Poll PSTAT.rd_port_busy_n = 0 Poll PSTAT.wr_port_busy_n = 0	Waits unit all AXI ports are idle
3	Write 1 to PWRCTL.selfref_sw	Causes system to move to Self-refresh state
4	Poll STAT.selfref_type= 2’b10	Waits until Self-refresh state is entered
5	Place IOs in Retention mode	Refer to the section “DDR/LPDDR Physical Interface (DDR3PHY)”
6	Remove power	–

Table 17-7. Re-enabling the Power

Step	Description	Comment
1	Enable Power	–
2	Reset controller/PHY by driving core_ddrc_rstn = 1’b0, aresetn_n = 1’b0, presetn = ’b0	_
3	Remove APB reset, presetn =1’b1, and rereprogram the registers to pre-power removal values	_
4	Program INIT0.skip_dram_init = 2’b11	Skips the DRAM init routine and starts up in Self-refresh mode
5	Programs PWRCTL.selfref_sw = 1’b1	Keeps the controller in Self-refresh mode
6	Program DFIMISC.dfi_init_complete_en to 1’b0	PHY initialization needs to be rerun, so set to 0 until initialization complete
7	Remove the core reset core_ddrc_rstn = 1’b1 aresetn_n = 1’b1	–
8	Run PHY initialization/training as required, including removing the IOs from Retention mode	Refer to the section “DDR3PHY”
9	Program DFIMISC.dfi_init_complete_en to 1’b1	Indicates to controller that PHY has completed re-training/initialization
10	Program PWRCTL.selfref_sw = 1’b0	Triggers Self-refresh exit
11	Poll STAT.selfref_type = 2’b00	Wait until Self-refresh state is exited
12	Poll STAT.operating_mode for Normal mode entry	–
13	Write PCTRL.port_en = 1	AXI ports are no longer blocked from taking transactions

17.5.14.5 Registers Related to Power-Saving Features

The following registers are related to the power saving features:

- UDDRC_DRAMTMG5.t_cke
- Precharge power-down controls:
  - UDDRC_PWRCTL.POWERDOWN_EN
  - UDDRC_PWRTMG.POWERDOWN_TO_X32
  - UDDRC_DRAMTMG1.T_XP
- Self-refresh controls:
  - UDDRC_PWRCTL.SELFREF_SW
  - UDDRC_PWRCTL.SELFREF_EN
  - UDDRC_HWLPCTL.HW_LP_EN
- Hardware low-power interface: csysreq_*/csysack_*/cactive_*
  - UDDRC_PWRTMG.SELFREF_TO_x32
  - UDDRC_STAT.SELFREF_TYPE
  - UDDRC_PWRCTL.STAY_IN_SELFREF
  - UDDRC_STAT.SELFREF_STATE
- Hardware low-power interface:
  - UDDRC_HWLPCTL.HW_LP_EXIT_IDLE_EN
  - UDDRC_HWLPCTL.HW_LP_IDLE_X32
- Deep power-down controls:
  - UDDRC_PWRCTL.DEEPPOWERDOWN_EN
  - UDDRC_PWRTMG.T_DPD_X4096

- Maximum Power-saving mode controls:
  - UDDRC_PWRCTL.MPSM_EN
  - UDDRC_DRAMTMG11.T_MPX_LH
  - UDDRC_DRAMTMG11.T_MPX_S
  - UDDRC_DRAMTMG11.T_CKMPE
- Assertion of dfi_dram_clk_disable to disable the clocks to the DRAM controls:
  - UDDRC_WRCTL.EN_DFI_DRAM_CLK_DISABLE
  - UDDRC_DFITMG1.DFI_T_DRAM_CLK_DISABLE
  - UDDRC_DFITMG1.DFI_T_DRAM_CLK_ENABLE
  - UDDRC_DRAMTMG5.T_CKSRE
  - UDDRC_DRAMTMG5.T_CKSRX
  - UDDRC_DRAMTMG6.T_CKPDE
  - UDDRC_DRAMTMG6.T_CKPDX
  - UDDRC_RAMTMG6.T_CKDPDE
  - UDDRC_DRAMTMG6.T_CKDPDX
  - UDDRC_DRAMTMG6.T_CKCSX

For more information about these registers, see Register Descriptions.

17.6 Register Summary

In the following register descriptions, "Programming Mode" specifies the register type:

- Static: can be written only when the controller is in reset.
- Dynamic: can be written at any time during operation.
- Quasi-dynamic: can be written when the controller is in reset and some specific conditions outside reset. There are four groups this type.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	UDDRC_MSTR	31:24								
		23:16					BURST_RDWR[3:0]			
		15:8	DLL_OFF_MODE		DATA_BUS_WIDTH[1:0]			EN_2T_TIMING_MODE	BURSTCHOP	
		7:0					LPDDR3	LPDDR2		DDR3
0x04	UDDRC_STAT	31:24								
		23:16								
		15:8				SELFREF_CAM_NOT_EMPTY				
		7:0			SELFREF_TYPE[1:0]			OPERATING_MODE[2:0]		
0x08 ... 0x0F	Reserved									
0x10	UDDRC_MRCTRL0	31:24	MR_WR							
		23:16								
		15:8	MR_ADDR[3:0]							
		7:0				MR_RANK				MR_TYPE
0x14	UDDRC_MRCTRL1	31:24								
		23:16								
		15:8	MR_DATA[15:8]							
		7:0	MR_DATA[7:0]							
0x18	UDDRC_MRSTAT	31:24								
		23:16								
		15:8								
		7:0								MR_WR_BUSY
0x1C ... 0x1F	Reserved									
0x20	UDDRC_DERATEEN	31:24								
		23:16								
		15:8				DERATE_MR4_TUF_DIS				
		7:0	DERATE_BYTE[3:0]					DERATE_VALUE[1:0]		DERATE_ENABLE
0x24	UDDRC_DERATEINT	31:24	MR4_READ_INTERVAL[31:24]							
		23:16	MR4_READ_INTERVAL[23:16]							
		15:8	MR4_READ_INTERVAL[15:8]							
		7:0	MR4_READ_INTERVAL[7:0]							
0x28 ... 0x2B	Reserved									
0x2C	UDDRC_DERATECTL	31:24								
		23:16								
		15:8								
		7:0						DERATE_TEMP_LIMIT_INTR_FORCE	DERATE_TEMP_LIMIT_INTR_CLR	DERATE_TEMP_LIMIT_INTR_EN

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	UDDRC_PWRCTL	31:24								
		23:16								
		15:8								
		7:0	DIS_CAM_DRAIN_SELFREF		SELFREF_SW		EN_DFI_DRAM_CLK_DISABLE	DEEPPOWERDOWN_EN	POWERDOWN_EN	SELFREF_EN
0x34	UDDRC_PWRTMG	31:24								
		23:16	SELFREF_TO_X32[7:0]							
		15:8	T_DPD_X4096[7:0]							
		7:0				POWERDOWN_TO_X32[4:0]				
0x38	UDDRC_HWLPCTL	31:24					HW_LP_IDLE_X32[11:8]			
		23:16	HW_LP_IDLE_X32[7:0]							
		15:8								
		7:0							HW_LP_EXIT_IDLE_EN	HW_LP_EN
0x3C ... 0x4F	Reserved									
0x50	UDDRC_RFSHCTL0	31:24								
		23:16	REFRESH_MARGIN[3:0]							REFRESH_TO_X1_X32[4]
		15:8	REFRESH_TO_X1_X32[3:0]						REFRESH_BURST[5:4]	
		7:0	REFRESH_BURST[3:0]					PER_BANK_REFRESH		
0x54 ... 0x5F	Reserved									
0x60	UDDRC_RFSHCTL3	31:24								
		23:16								
		15:8								
		7:0							REFRESH_UPDATE_LEVEL	DIS_AUTO_REFRESH
0x64	UDDRC_RFSHTMG	31:24	T_RFC_NOM_X1_SEL				T_RFC_NOM_X1_X32[11:8]			
		23:16	T_RFC_NOM_X1_X32[7:0]							
		15:8	LPDDR3_TREFBW_EN						T_RFC_MIN[9:8]	
		7:0	T_RFC_MIN[7:0]							
0x68 ... 0xBF	Reserved									
0xC0	UDDRC_CRCPARCTL0	31:24								
		23:16								
		15:8								
		7:0						DFI_ALERT_ERR_CNT_CLR	DFI_ALERT_ERR_INT_CLR	DFI_ALERT_ERR_INT_EN

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC4 ... 0xCB	Reserved									
0xCC	UDDRC_CRCPARSTAT	31:24								
		23:16								DFI_ALERT_ERR_INT
		15:8	DFI_ALERT_ERR_CNT[15:8]							
		7:0	DFI_ALERT_ERR_CNT[7:0]							
0xD0	UDDRC_INIT0	31:24	SKIP_DRAM_INIT[1:0]						POST_CKE_X1024[9:8]	
		23:16	POST_CKE_X1024[7:0]							
		15:8					PRE_CKE_X1024[11:8]			
		7:0	PRE_CKE_X1024[7:0]							
0xD4	UDDRC_INIT1	31:24								DRAM_RSTN_X1024[8]
		23:16	DRAM_RSTN_X1024[7:0]							
		15:8								
		7:0					PRE_OCD_X32[3:0]			
0xD8	UDDRC_INIT2	31:24								
		23:16								
		15:8	IDLE_AFTER_RESET_X32[7:0]							
		7:0					MIN_STABLE_CLOCK_X1[3:0]			
0xDC	UDDRC_INIT3	31:24	MR[15:8]							
		23:16	MR[7:0]							
		15:8	EMR[15:8]							
		7:0	EMR[7:0]							
0xE0	UDDRC_INIT4	31:24	EMR2[15:8]							
		23:16	EMR2[7:0]							
		15:8	EMR3[15:8]							
		7:0	EMR3[7:0]							
0xE4	UDDRC_INIT5	31:24								
		23:16	DEV_ZQINIT_X32[7:0]							
		15:8							MAX_AUTO_INIT_X1024[9:8]	
		7:0	MAX_AUTO_INIT_X1024[7:0]							
0xE8 ... 0xEF	Reserved									
0xF0	UDDRC_DIMMCTL	31:24								
		23:16								
		15:8								
		7:0							DIMM_ADDR_MIRR_EN	DIMM_STAGGER_CS_EN
0xF4 ... 0xFF	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100	UDDRC_DRAMTMG0	31:24		WR2PRE[6:0]						
		23:16			T_FAW[5:0]					
		15:8		T_RAS_MAX[6:0]						
		7:0			T_RAS_MIN[5:0]					
0x0104	UDDRC_DRAMTMG1	31:24								
		23:16				T_XP[4:0]				
		15:8			RD2PRE[5:0]					
		7:0		T_RC[6:0]						
0x0108	UDDRC_DRAMTMG2	31:24			WRITE_LATENCY[5:0]					
		23:16			READ_LATENCY[5:0]					
		15:8			RD2WR[5:0]					
		7:0			WR2RD[5:0]					
0x010C	UDDRC_DRAMTMG3	31:24			T_MRW[9:4]					
		23:16	T_MRW[3:0]						T_MRD[5:4]	
		15:8	T_MRD[3:0]						T_MOD[9:8]	
		7:0	T_MOD[7:0]							
0x0110	UDDRC_DRAMTMG4	31:24				T_RCD[4:0]				
		23:16					T_CCD[3:0]			
		15:8					T_RRD[3:0]			
		7:0				T_RP[4:0]				
0x0114	UDDRC_DRAMTMG5	31:24					T_CKSRX[3:0]			
		23:16					T_CKSRE[3:0]			
		15:8			T_CKESR[5:0]					
		7:0				T_CKE[4:0]				
0x0118	UDDRC_DRAMTMG6	31:24					T_CKDPDE[3:0]			
		23:16					T_CKDPDX[3:0]			
		15:8								
		7:0					T_CKCSX[3:0]			
0x011C	UDDRC_DRAMTMG7	31:24								
		23:16								
		15:8					T_CKPDE[3:0]			
		7:0					T_CKPDX[3:0]			
0x0120	UDDRC_DRAMTMG8	31:24								
		23:16								
		15:8		T_XS_DLL_X32[6:0]						
		7:0		T_XS_X32[6:0]						
0x0124 ... 0x0137	Reserved									
0x0138	UDDRC_DRAMTMG14	31:24								
		23:16								
		15:8					T_XSR[11:8]			
		7:0	T_XSR[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x013C	UDDRC_DRAMTMG15	31:24	EN_DFI_LP_T_STAB							
		23:16								
		15:8								
		7:0	T_STAB_X32[7:0]							
0x0140 ... 0x017F	Reserved									
0x0180	UDDRC_ZQCTL0	31:24	DIS_AUTO_ZQ	DIS_SRX_ZQCL	ZQ_RESISTOR_SHARED			T_ZQ_LONG_NOP[10:8]		
		23:16	T_ZQ_LONG_NOP[7:0]							
		15:8							T_ZQ_SHORT_NOP[9:8]	
		7:0	T_ZQ_SHORT_NOP[7:0]							
0x0184	UDDRC_ZQCTL1	31:24			T_ZQ_RESET_NOP[9:4]					
		23:16	T_ZQ_RESET_NOP[3:0]				T_ZQ_SHORT_INTERVAL_X1024[19:16]			
		15:8	T_ZQ_SHORT_INTERVAL_X1024[15:8]							
		7:0	T_ZQ_SHORT_INTERVAL_X1024[7:0]							
0x0188	UDDRC_ZQCTL2	31:24								
		23:16								
		15:8								
		7:0								ZQ_RESET
0x018C	UDDRC_ZQSTAT	31:24								
		23:16								
		15:8								
		7:0								ZQ_RESET_BUSY
0x0190	UDDRC_DFITMG0	31:24				DFI_T_CTRL_DELAY[4:0]				
		23:16	DFI_RDDATA_USE_DFI_PHY_CLK	DFI_T_RDDATA_EN[6:0]						
		15:8	DFI_WRDATA_USE_DFI_PHY_CLK		DFI_TPHY_WRDATA[5:0]					
		7:0			DFI_TPHY_WRLAT[5:0]					
0x0194	UDDRC_DFITMG1	31:24							DFI_T_PARIN_LAT[1:0]	
		23:16				DFI_T_WRDATA_DELAY[4:0]				
		15:8				DFI_T_DRAM_CLK_DISABLE[4:0]				
		7:0				DFI_T_DRAM_CLK_ENABLE[4:0]				
0x0198	UDDRC_DFILPCFG0	31:24				DFI_TLP_RESP[4:0]				
		23:16	DFI_LP_WAKEUP_DPD[3:0]							DFI_LP_EN_DPD
		15:8	DFI_LP_WAKEUP_SR[3:0]							DFI_LP_EN_SR
		7:0	DFI_LP_WAKEUP_PD[3:0]							DFI_LP_EN_PD
0x019C ... 0x019F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01A0	UDDRC_DFIUPD0	31:24	DIS_AUTO_CTRLUPD	DIS_AUTO_CTRLUPD_SRX	CTRLUPD_PRE_SRX				DFI_T_CTRLUP_MAX[9:8]	
		23:16	DFI_T_CTRLUP_MAX[7:0]							
		15:8							DFI_T_CTRLUP_MIN[9:8]	
		7:0	DFI_T_CTRLUP_MIN[7:0]							
0x01A4	UDDRC_DFIUPD1	31:24								
		23:16	DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0]							
		15:8								
		7:0	DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0]							
0x01A8	UDDRC_DFIUPD2	31:24	DFI_PHYUPD_EN							
		23:16								
		15:8								
		7:0								
0x01AC ... 0x01AF	Reserved									
0x01B0	UDDRC_DFIMISC	31:24								
		23:16								
		15:8				DFI_FREQUENCY[4:0]				
		7:0			DFI_INIT_START	CTL_IDLE_EN				DFI_INIT_COMPLETE_EN
0x01B4 ... 0x01BB	Reserved									
0x01BC	UDDRC_DFISTAT	31:24								
		23:16								
		15:8								
		7:0							DFI_LP_ACK	DFI_INIT_COMPLETE
0x01C0 ... 0x01C3	Reserved									
0x01C4	UDDRC_DFIPHYMSTR	31:24								
		23:16								
		15:8								
		7:0								DFI_PHYMSTR_EN
0x01C8 ... 0x0203	Reserved									
0x0204	UDDRC_ADDRMAP1	31:24								
		23:16			ADDRMAP_BANK_B2[5:0]					
		15:8			ADDRMAP_BANK_B1[5:0]					
		7:0			ADDRMAP_BANK_B0[5:0]					

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0208	UDDRC_ADDRMAP2	31:24					ADDRMAP_COL_B5[3:0]			
		23:16					ADDRMAP_COL_B4[3:0]			
		15:8				ADDRMAP_COL_B3[4:0]				
		7:0					ADDRMAP_COL_B2[3:0]			
0x020C	UDDRC_ADDRMAP3	31:24				ADDRMAP_COL_B9[4:0]				
		23:16				ADDRMAP_COL_B8[4:0]				
		15:8				ADDRMAP_COL_B7[4:0]				
		7:0				ADDRMAP_COL_B6[4:0]				
0x0210	UDDRC_ADDRMAP4	31:24								
		23:16								
		15:8				ADDRMAP_COL_B11[4:0]				
		7:0				ADDRMAP_COL_B10[4:0]				
0x0214	UDDRC_ADDRMAP5	31:24					ADDRMAP_ROW_B11[3:0]			
		23:16					ADDRMAP_ROW_B2_10[3:0]			
		15:8					ADDRMAP_ROW_B1[3:0]			
		7:0					ADDRMAP_ROW_B0[3:0]			
0x0218	UDDRC_ADDRMAP6	31:24	LPDDR3_6GB_12GB				ADDRMAP_ROW_B15[3:0]			
		23:16					ADDRMAP_ROW_B14[3:0]			
		15:8					ADDRMAP_ROW_B13[3:0]			
		7:0					ADDRMAP_ROW_B12[3:0]			
0x021C ... 0x0223	Reserved									
0x0224	UDDRC_ADDRMAP9	31:24					ADDRMAP_ROW_B5[3:0]			
		23:16					ADDRMAP_ROW_B4[3:0]			
		15:8					ADDRMAP_ROW_B3[3:0]			
		7:0					ADDRMAP_ROW_B2[3:0]			
0x0228	UDDRC_ADDRMAP10	31:24					ADDRMAP_ROW_B9[3:0]			
		23:16					ADDRMAP_ROW_B8[3:0]			
		15:8					ADDRMAP_ROW_B7[3:0]			
		7:0					ADDRMAP_ROW_B6[3:0]			
0x022C	UDDRC_ADDRMAP11	31:24								
		23:16								
		15:8								
		7:0					ADDRMAP_ROW_B10[3:0]			
0x0230 ... 0x023F	Reserved									
0x0240	UDDRC_ODTCFG	31:24					WR_ODT_HOLD[3:0]			
		23:16				WR_ODT_DELAY[4:0]				
		15:8					RD_ODT_HOLD[3:0]			
		7:0		RD_ODT_DELAY[4:0]						

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0244	UDDRC_ODTMAP	31:24								
		23:16								
		15:8								
		7:0				RANK0_RD_ODT				RANK0_WR_ODT
0x0248 ... 0x024F	Reserved									
0x0250	UDDRC_SCHED	31:24		RDWR_IDLE_GAP[6:0]						
		23:16	GO2CRITICAL_HYSTERESIS[7:0]							
		15:8				LPR_NUM_ENTRIES[4:0]				
		7:0						PAGECLOSE	PREFER_WRITE	FORCE_LOW_PRI_N
0x0254	UDDRC_SCHED1	31:24								
		23:16								
		15:8								
		7:0	PAGECLOSE_TIMER[7:0]							
0x0258 ... 0x025B	Reserved									
0x025C	UDDRC_PERFHPR1	31:24	HPR_XACT_RUN_LENGTH[7:0]							
		23:16								
		15:8	HPR_MAX_STARVE[15:8]							
		7:0	HPR_MAX_STARVE[7:0]							
0x0260 ... 0x0263	Reserved									
0x0264	UDDRC_PERFLPR1	31:24	LPR_XACT_RUN_LENGTH[7:0]							
		23:16								
		15:8	LPR_MAX_STARVE[15:8]							
		7:0	LPR_MAX_STARVE[7:0]							
0x0268 ... 0x026B	Reserved									
0x026C	UDDRC_PERFWR1	31:24	W_XACT_RUN_LENGTH[7:0]							
		23:16								
		15:8	W_MAX_STARVE[15:8]							
		7:0	W_MAX_STARVE[7:0]							
0x0270 ... 0x02FF	Reserved									
0x0300	UDDRC_DBG0	31:24								
		23:16								
		15:8								
		7:0				DIS_COLLISION_PAGE_OPT		DIS_ACT_BYPASS	DIS_RD_BYPASS	DIS_WC

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0304	UDDRC_DBG1	31:24								
		23:16								
		15:8								
		7:0							DIS_HIF	DIS_DQ
0x0308	UDDRC_DBGCAM	31:24			WR_DATA_PIPELINE_EMPTY	RD_DATA_PIPELINE_EMPTY		DBG_WR_Q_EMPTY	DBG_RD_Q_EMPTY	DBG_STALL
		23:16			DBG_W_Q_DEPTH[5:0]					
		15:8			DBG_LPR_Q_DEPTH[5:0]					
		7:0			DBG_HPR_Q_DEPTH[5:0]					
0x030C	UDDRC_DBGCMD	31:24								
		23:16								
		15:8								
		7:0			CTRLUPD	ZQ_CALIB_SHORT				RANK0_REFRESH
0x0310	UDDRC_DBGSTAT	31:24								
		23:16								
		15:8								
		7:0			CTRLUPD_BUSY	ZQ_CALIB_SHORT_BUSY				RANK0_REFRESH_BUSY
0x0314 ... 0x031F	Reserved									
0x0320	UDDRC_SWCTL	31:24								
		23:16								
		15:8								
		7:0								SW_DONE
0x0324	UDDRC_SWSTAT	31:24								
		23:16								
		15:8								
		7:0								SW_DONE_ACK
0x0328 ... 0x036B	Reserved									
0x036C	UDDRC_POISONCFG	31:24								RD_POISON_INTR_CLR
		23:16				RD_POISON_INTR_EN				RD_POISON_SLVERR_EN
		15:8								WR_POISON_INTR_CLR
		7:0				WR_POISON_INTR_EN				WR_POISON_SLVERR_EN

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0370	UDDRC_POISONSTAT	31:24								
		23:16				RD_POISON_INTR_4	RD_POISON_INTR_3	RD_POISON_INTR_2	RD_POISON_INTR_1	RD_POISON_INTR_0
		15:8								
		7:0				WR_POISON_INTR_4	WR_POISON_INTR_3	WR_POISON_INTR_2	WR_POISON_INTR_1	WR_POISON_INTR_0
0x0374 ... 0x03EF	Reserved									
0x03F0	UDDRC_DERATESTAT	31:24								
		23:16								
		15:8								
		7:0								DERATE_TEMP_LIMIT_INTR
0x03F4 ... 0x03FB	Reserved									
0x03FC	UDDRC_PSTAT	31:24								
		23:16				WR_PORT_BUSY_4	WR_PORT_BUSY_3	WR_PORT_BUSY_2	WR_PORT_BUSY_1	WR_PORT_BUSY_0
		15:8								
		7:0				RD_PORT_BUSY_4	RD_PORT_BUSY_3	RD_PORT_BUSY_2	RD_PORT_BUSY_1	RD_PORT_BUSY_0
0x0400	UDDRC_PCCFG	31:24								
		23:16								
		15:8								BL_EXP_MODE
		7:0				PAGEMATCH_LIMIT				GO2CRITICAL_EN
0x0404	UDDRC_PCFGR_0	31:24								
		23:16								
		15:8		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
		7:0	RD_PORT_PRIORITY[7:0]							
0x0408	UDDRC_PCFGW_0	31:24								
		23:16								
		15:8		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
		7:0	WR_PORT_PRIORITY[7:0]							
0x040C ... 0x048F	Reserved									
0x0490	UDDRC_PCTRL_0	31:24								
		23:16								
		15:8								
		7:0								PORT_EN

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0494	UDDRC_PCFGQOS0_0	31:24								
		23:16			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
		15:8								
		7:0					RQOS_MAP_LEVEL1[3:0]			
0x0498	UDDRC_PCFGQOS1_0	31:24						RQOS_MAP_TIMEOUTR[10:8]		
		23:16	RQOS_MAP_TIMEOUTR[7:0]							
		15:8						RQOS_MAP_TIMEOUTB[10:8]		
		7:0	RQOS_MAP_TIMEOUTB[7:0]							
0x049C	UDDRC_PCFGWQOS0_0	31:24							WQOS_MAP_REGION2[1:0]	
		23:16			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
		15:8					WQOS_MAP_LEVEL2[3:0]			
		7:0					WQOS_MAP_LEVEL1[3:0]			
0x04A0	UDDRC_PCFGWQOS1_0	31:24						WQOS_MAP_TIMEOUT2[10:8]		
		23:16	WQOS_MAP_TIMEOUT2[7:0]							
		15:8						WQOS_MAP_TIMEOUT1[10:8]		
		7:0	WQOS_MAP_TIMEOUT1[7:0]							
0x04A4 ... 0x04B3	Reserved									
0x04B4	UDDRC_PCFGR_1	31:24								
		23:16								
		15:8		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
		7:0	RD_PORT_PRIORITY[7:0]							
0x04B8	UDDRC_PCFGW_1	31:24								
		23:16								
		15:8		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
		7:0	WR_PORT_PRIORITY[7:0]							
0x04BC ... 0x053F	Reserved									
0x0540	UDDRC_PCTRL_1	31:24								
		23:16								
		15:8								
		7:0								PORT_EN
0x0544	UDDRC_PCFGQOS0_1	31:24								
		23:16			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
		15:8								
		7:0					RQOS_MAP_LEVEL1[3:0]			
0x0548	UDDRC_PCFGQOS1_1	31:24						RQOS_MAP_TIMEOUTR[10:8]		
		23:16	RQOS_MAP_TIMEOUTR[7:0]							
		15:8						RQOS_MAP_TIMEOUTB[10:8]		
		7:0	RQOS_MAP_TIMEOUTB[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x054C	UDDRC_PCFGWQOS0_1	31:24							WQOS_MAP_REGION2[1:0]	
		23:16			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
		15:8					WQOS_MAP_LEVEL2[3:0]			
		7:0					WQOS_MAP_LEVEL1[3:0]			
0x0550	UDDRC_PCFGWQOS1_1	31:24						WQOS_MAP_TIMEOUT2[10:8]		
		23:16	WQOS_MAP_TIMEOUT2[7:0]							
		15:8						WQOS_MAP_TIMEOUT1[10:8]		
		7:0	WQOS_MAP_TIMEOUT1[7:0]							
0x0554 ... 0x0563	Reserved									
0x0564	UDDRC_PCFGR_2	31:24								
		23:16								
		15:8		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
		7:0	RD_PORT_PRIORITY[7:0]							
0x0568	UDDRC_PCFGW_2	31:24								
		23:16								
		15:8		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
		7:0	WR_PORT_PRIORITY[7:0]							
0x056C ... 0x05EF	Reserved									
0x05F0	UDDRC_PCTRL_2	31:24								
		23:16								
		15:8								
		7:0								PORT_EN
0x05F4	UDDRC_PCFGQOS0_2	31:24							RQOS_MAP_REGION2[1:0]	
		23:16			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
		15:8					RQOS_MAP_LEVEL2[3:0]			
		7:0					RQOS_MAP_LEVEL1[3:0]			
0x05F8	UDDRC_PCFGQOS1_2	31:24						RQOS_MAP_TIMEOUTR[10:8]		
		23:16	RQOS_MAP_TIMEOUTR[7:0]							
		15:8						RQOS_MAP_TIMEOUTB[10:8]		
		7:0	RQOS_MAP_TIMEOUTB[7:0]							
0x05FC	UDDRC_PCFGWQOS0_2	31:24							WQOS_MAP_REGION2[1:0]	
		23:16			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
		15:8					WQOS_MAP_LEVEL2[3:0]			
		7:0					WQOS_MAP_LEVEL1[3:0]			
0x0600	UDDRC_PCFGWQOS1_2	31:24						WQOS_MAP_TIMEOUT2[10:8]		
		23:16	WQOS_MAP_TIMEOUT2[7:0]							
		15:8						WQOS_MAP_TIMEOUT1[10:8]		
		7:0	WQOS_MAP_TIMEOUT1[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0604 ... 0x0613	Reserved									
0x0614	UDDRC_PCFGR_3	31:24								
		23:16								
		15:8		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
		7:0	RD_PORT_PRIORITY[7:0]							
0x0618	UDDRC_PCFGW_3	31:24								
		23:16								
		15:8		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
		7:0	WR_PORT_PRIORITY[7:0]							
0x061C ... 0x069F	Reserved									
0x06A0	UDDRC_PCTRL_3	31:24								
		23:16								
		15:8								
		7:0								PORT_EN
0x06A4	UDDRC_PCFGQOS0_3	31:24								
		23:16			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
		15:8								
		7:0					RQOS_MAP_LEVEL1[3:0]			
0x06A8	UDDRC_PCFGQOS1_3	31:24						RQOS_MAP_TIMEOUTR[10:8]		
		23:16	RQOS_MAP_TIMEOUTR[7:0]							
		15:8						RQOS_MAP_TIMEOUTB[10:8]		
		7:0	RQOS_MAP_TIMEOUTB[7:0]							
0x06AC	UDDRC_PCFGWQOS0_3	31:24							WQOS_MAP_REGION2[1:0]	
		23:16			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
		15:8					WQOS_MAP_LEVEL2[3:0]			
		7:0					WQOS_MAP_LEVEL1[3:0]			
0x06B0	UDDRC_PCFGWQOS1_3	31:24						WQOS_MAP_TIMEOUT2[10:8]		
		23:16	WQOS_MAP_TIMEOUT2[7:0]							
		15:8						WQOS_MAP_TIMEOUT1[10:8]		
		7:0	WQOS_MAP_TIMEOUT1[7:0]							
0x06B4 ... 0x06C3	Reserved									
0x06C4	UDDRC_PCFGR_4	31:24								
		23:16								
		15:8		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
		7:0	RD_PORT_PRIORITY[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x06C8	UDDRC_PCFGW_4	31:24								
		23:16								
		15:8		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
		7:0	WR_PORT_PRIORITY[7:0]							
0x06CC ... 0x074F	Reserved									
0x0750	UDDRC_PCTRL_4	31:24								
		23:16								
		15:8								
		7:0								PORT_EN
0x0754	UDDRC_PCFGQOS0_4	31:24								
		23:16			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
		15:8								
		7:0					RQOS_MAP_LEVEL1[3:0]			
0x0758	UDDRC_PCFGQOS1_4	31:24						RQOS_MAP_TIMEOUTR[10:8]		
		23:16	RQOS_MAP_TIMEOUTR[7:0]							
		15:8						RQOS_MAP_TIMEOUTB[10:8]		
		7:0	RQOS_MAP_TIMEOUTB[7:0]							
0x075C	UDDRC_PCFGWQOS0_4	31:24							WQOS_MAP_REGION2[1:0]	
		23:16			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
		15:8					WQOS_MAP_LEVEL2[3:0]			
		7:0					WQOS_MAP_LEVEL1[3:0]			
0x0760	UDDRC_PCFGWQOS1_4	31:24						WQOS_MAP_TIMEOUT2[10:8]		
		23:16	WQOS_MAP_TIMEOUT2[7:0]							
		15:8						WQOS_MAP_TIMEOUT1[10:8]		
		7:0	WQOS_MAP_TIMEOUT1[7:0]							
0x0764 ... 0x0F03	Reserved									
0x0F04	UDDRC_SARBASE0	31:24								
		23:16								
		15:8								
		7:0						BASE_ADDR[2:0]		
0x0F08	UDDRC_SARSIZE0	31:24								
		23:16								
		15:8								
		7:0	NBLOCKS[7:0]							

17.6.1 UDDRC Host Register 0

Name: UDDRC_MSTR
Offset: 0x000
Reset: 0x00040001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					BURST_RDWR[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	15	14	13	12	11	10	9	8
	DLL_OFF_MODE		DATA_BUS_WIDTH[1:0]			EN_2T_TIMING_MODE	BURSTCHOP	
Access	R/W		R/W	R/W		R/W	R/W	
Reset	0		0	0		0	0	

Bit	7	6	5	4	3	2	1	0
					LPDDR3	LPDDR2		DDR3
Access					R/W	R/W		R/W
Reset					0	0		1

Bits 19:16 – BURST_RDWR[3:0] SDRAM burst length used
This controls the burst size used to access the SDRAM. This must match the burst length mode register setting in the SDRAM. (For BC4/8 on-the-fly mode of DDR3, set this field to 0x0100). Burst length of 2 is not supported.
Burst length of 2 is only supported when the controller is operating in 1:1 frequency mode.
For DDR3 and LPDDR3, this must be set to 0x0100 (BL8).
Programming mode: Static

Value	Description
0001	Reserved
0010	Burst length of 4
0100	Burst length of 8
1000	Burst length of 16 (only supported for LPDDR2)
–	All other values are reserved.

Bit 15 – DLL_OFF_MODE
Set to 1 when the UDDRC and DRAM has to be put in DLL-off mode for low frequency operation.
Set to 0 to put UDDRC and DRAM in DLL-on mode for normal frequency operation.
Programming mode: Quasi-dynamic Group 2

Bits 13:12 – DATA_BUS_WIDTH[1:0]
Selects proportion of DQ bus width that is used by the SDRAM.
Note that half bus width mode is only supported when the SDRAM bus width is a multiple of 16. Bus width refers to DQ bus width.
Programming mode: Static

Value	Description
00	Full DQ bus width to SDRAM
01	Half DQ bus width to SDRAM
10	Quarter DQ bus width to SDRAM
11	Reserved

Bit 10 – EN_2T_TIMING_MODE
If 1, then UDDRC uses 2T timing. Otherwise, uses 1T timing. In 2T timing, all command signals (except chip select) are held for 2 clocks on the SDRAM bus. Chip select is asserted on the second cycle of the command.
2T timing is not supported in the following cases:

- in LPDDR2/LPDDR3 mode
- in Shared-AC dual channel mode and the register value is don't care.

Programming mode: Static

Bit 9 – BURSTCHOP
When set, enable burst-chop (BC4 or 8 on-the-fly) in DDR3. Burst Chop for Reads is not exercised. Burst Chop for Writes is exercised.
BC4 (fixed) mode is not supported.
Programming mode: Static

Bit 3 – LPDDR3 Select LPDDR3 SDRAM
Present only in designs configured to support LPDDR3.
Programming mode: Static

Value	Description
0	Non-LPDDR3 device in use.
1	LPDDR3 SDRAM device in use.

Bit 2 – LPDDR2 Select LPDDR2 SDRAM
Present only in designs configured to support LPDDR2.
Programming mode: Static

Value	Description
0	Non-LPDDR2 device in use
1	LPDDR2 SDRAM device in use.

Bit 0 – DDR3 Select DDR3 SDRAM
Only present in designs that support DDR3.
Programming mode: Static

Value	Description
0	Non-DDR3 SDRAM device in use
1	DDR3 SDRAM device in use

17.6.2 UDDRC Operating Mode Status Register

Name: UDDRC_STAT
Offset: 0x004
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SELFREF_CAM_NOT_EMPTY				
Access				R				
Reset				0				

Bit	7	6	5	4	3	2	1	0
			SELFREF_TYPE[1:0]			OPERATING_MODE[2:0]		
Access			R	R		R	R	R
Reset			0	0		0	0	0

Bit 12 – SELFREF_CAM_NOT_EMPTY Self-refresh with CAMs not empty
Set to 1 when Self-refresh is entered but CAMs are not drained. Cleared after exiting Self-refresh.
Programming Mode: Static

Bits 5:4 – SELFREF_TYPE[1:0]
Flags if Self-refresh is entered and if it was under Automatic Self-refresh control only or not.
Programming Mode: Static

Value	Description
00	SDRAM is not in Self-refresh. If retry is enabled by CRCPARCTRL1.crc_parity_retry_enable, this also indicates the SRE command is still in parity error window or retry is in progress.
11	SDRAM is in Self-refresh, which was caused by Automatic Self-refresh only. If retry is enabled, this ensures the SRE command is executed correctly without parity error.
10	SDRAM is in Self-refresh, which was not caused solely under Automatic Self-refresh control. It could have been caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw). If retry is enabled, this ensures the SRE command is executed correctly without parity error.
01	SDRAM is in Self-refresh, which was caused by PHY Host Request.

Bits 2:0 – OPERATING_MODE[2:0] Operating mode
This is 3-bit wide in configurations with LPDDR2/LPDDR3 support and 2-bit wide in all other configurations.
Programming Mode: Static

Value	Description
	Non-LPDDR2/LPDDR3 designs:
00	Init
01	Normal
10	Power-down
11	Self-refresh
	LPDDR2/LPDDR3 designs:

Value	Description
000	Init
001	Normal
010	Power-down
011	Self-refresh
1XX	Deep Power-down / Maximum Power Saving mode

17.6.3 UDDRC Mode Register Read/Write Control Register 0

Name: UDDRC_MRCTRL0
Offset: 0x010
Reset: 0x00000010
Property: R/W

Bit	31	30	29	28	27	26	25	24
	MR_WR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MR_ADDR[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
				MR_RANK				MR_TYPE
Access				R/W				R/W
Reset				1				0

Bit 31 – MR_WR
Setting this register bit to 1 triggers a mode register read or write operation. When the MR operation is complete, the UDDRC automatically clears this bit.
The other register fields of this register must be written in a separate APB transaction, before setting this mr_wr bit. It is recommended NOT to set this signal if in Init, Deep Power-down or MPSM operating modes.
Programming mode: Dynamic

Bits 15:12 – MR_ADDR[3:0] Address of the Mode register that is to be written to.
Don't Care for LPDDR2/LPDDR3 (see UDDRC_MRCTRL1.MR_DATA for mode register addressing in LPDDR2/LPDDR3).
This signal is also used for writing to control words of the register chip on RDIMMs/LRDIMMs. In that case, it corresponds to the bank address bits sent to the RDIMM/LRDIMM.
Programming mode: Dynamic

Value	Description
0000	MR0
0001	MR1
0010	MR2
0011	MR3
0100	MR4
0101	MR5
0110	MR6
0111	MR7

Bit 4 – MR_RANK
Controls which rank is accessed by MRCTRL0.mr_wr. Normally, it is desired to access all ranks, so all bits should be set to 1. However, for multi-rank UDIMMs/RDIMMs/LRDIMMs which implement address mirroring, it may be necessary to access ranks individually.
Examples (assume UDDRC is configured for 4 ranks):
Programming mode: Dynamic

Value	Description
0x1	Select rank 0 only
0x2	Select rank 1 only
0x5	Select ranks 0 and 2
0xA	Select ranks 1 and 3
0xF	Select ranks 0, 1, 2 and 3

Bit 0 – MR_TYPE
Indicates whether the Mode register operation is read or write. Only used for LPDDR2/LPDDR3.
Programming mode: Dynamic

Value	Description
0	Write
1	Read

17.6.4 UDDRC Mode Register Read/Write Control Register 1

Name: UDDRC_MRCTRL1
Offset: 0x014
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MR_DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MR_DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – MR_DATA[15:0]
Mode register write data for all non-LPDDR2/non-LPDDR3 modes.
For LPDDR2/LPDDR3, MRCTRL1[15:0] are interpreted as
[15:8] MR Address
[7:0] MR data for writes, don't care for reads. This is 16-bit wide for all configurations.
Programming mode: Dynamic

17.6.5 UDDRC Mode Register Read/Write Status Register

Name: UDDRC_MRSTAT
Offset: 0x018
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								MR_WR_BUSY
Access								R
Reset								0

Bit 0 – MR_WR_BUSY
The SoC core may initiate a MR write operation only if this signal is low. This signal goes high in the clock after the UDDRC accepts the MRW/MRR request. It goes low when the MRW/MRR command is issued to the SDRAM. It is recommended not to perform MRW/MRR commands when 'MRSTAT.mr_wr_busy' is high.
Programming mode: Dynamic

Value	Description
0	Indicates that the SoC core can initiate a mode register write operation.
1	Indicates that mode register write operation is in progress.

17.6.6 UDDRC Temperature Derate Enable Register

Name: UDDRC_DERATEEN
Offset: 0x020
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DERATE_MR4_TUF_DIS				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
	DERATE_BYTE[3:0]					DERATE_VALUE[1:0]		DERATE_ENABLE
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 12 – DERATE_MR4_TUF_DIS
Disable use of MR4 TUF flag (MR4[7]) bit.
For LPDDR2 and LPDDR3, contact your memory vendor for recommended usage.
This affects both the periodic derate logic (DERATEEN.derate_enable) and the derate_temp_limit_intr.
Programming mode: Quasi-dynamic Group 2, Group 4

Value	Description
0	Use MR4 TUF flag (MR4[7]).
1	Do not use MR4 TUF Flag (MR4[7]).

Bits 7:4 – DERATE_BYTE[3:0] Derate byte
Present only in designs configured to support LPDDR2/LPDDR3.
Indicates which byte of the MRR data is used for derating. The maximum valid value is 1.
Programming mode: Static

Bits 2:1 – DERATE_VALUE[1:0] Derate value
Present only in designs configured to support LPDDR2/LPDDR3.
Set to 0 for all LPDDR2 speed grades as derating value of +1.875 ns is less than a core_ddrc_core_clk period.
For LPDDR3, if the period of core_ddrc_core_clk is less than 1.875ns, this register field should be set to 1; otherwise it should be set to 0.
Programming mode: Quasi-dynamic Group 2, Group 4

Value	Description
0	Derating uses +1
1	Derating uses +2

Bit 0 – DERATE_ENABLE Enables derating
Present only in designs configured to support LPDDR2/LPDDR3.
This field must be set to '0' for non-LPDDR2/LPDDR3 mode.
Programming mode: Dynamic

Value	Description
0	Timing parameter derating is disabled.
1	Timing parameter derating is enabled using MR4 read value.

17.6.7 UDDRC Temperature Derate Interval Register

Name: UDDRC_DERATEINT
Offset: 0x024
Reset: 0x00800000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MR4_READ_INTERVAL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MR4_READ_INTERVAL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MR4_READ_INTERVAL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MR4_READ_INTERVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MR4_READ_INTERVAL[31:0] Interval between two MR4 reads
Used to derate the timing parameters.
Present only in designs configured to support LPDDR2/LPDDR3. This register must not be set to zero.
Unit: DFI clock cycles.
Programming mode: Static

17.6.8 UDDRC Temperature Derate Control Register

Name: UDDRC_DERATECTL
Offset: 0x02C
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DERATE_TEMP_LIMIT_INTR_FORCE	DERATE_TEMP_LIMIT_INTR_CLR	DERATE_TEMP_LIMIT_INTR_EN
Access						R/W	R/W	R/W
Reset						0	0	1

Bit 2 – DERATE_TEMP_LIMIT_INTR_FORCE Interrupt force bit for derate_temp_limit_intr
Setting this register to 1 will cause the derate_temp_limit_intr output pin to be asserted.
At the end of the interrupt force operation, the UDDRC automatically clears this bit.
Programming mode: Dynamic

Bit 1 – DERATE_TEMP_LIMIT_INTR_CLR Interrupt clear bit for derate_temp_limit_intr
At the end of the interrupt clear operation, the UDDRC automatically clears this bit.
Programming mode: Dynamic

Bit 0 – DERATE_TEMP_LIMIT_INTR_EN Interrupt enable bit for derate_temp_limit_intr output pin
Programming mode: Dynamic

Value	Description
0	Enabled
1	Disabled

17.6.9 UDDRC Low Power Control Register

Name: UDDRC_PWRCTL
Offset: 0x030
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DIS_CAM_DRAIN_SELFREF		SELFREF_SW		EN_DFI_DRAM_CLK_DISABLE	DEEPPOWERDOWN_EN	POWERDOWN_EN	SELFREF_EN
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit 7 – DIS_CAM_DRAIN_SELFREF
Indicates whether skipping CAM draining is allowed when entering Self-refresh.
This register field cannot be modified while PWRCTL.selfref_sw=1.
Note: PWRCTL.dis_cam_drain_selfref=1 is unsupported in this release. PWRCTL.dis_cam_drain_selfref=0 is required.
Programming mode: Dynamic

Value	Description
0	CAMs must be empty before entering SR.
1	CAMs are not emptied before entering SR (unsupported).

Bit 5 – SELFREF_SW
A value of 1 to this register causes system to move to Self-refresh state immediately, as long as it is not in Init or DPD/MPSM operating_mode. This is referred to as software entry/exit to Self-refresh.
Programming mode: Dynamic

Value	Description
0	Software entry to Self-refresh
1	Software exit from Self-refresh

Bit 3 – EN_DFI_DRAM_CLK_DISABLE
Enable the assertion of dfi_dram_clk_disable whenever a clock is not required by the SDRAM.
If set to 0, dfi_dram_clk_disable is never asserted.
Assertion of dfi_dram_clk_disable is as follows:

- In DDR2/DDR3, can only be asserted in Self-refresh.
- In LPDDR2/LPDDR3, can be asserted in the following cases:
  - in Self-refresh
  - in Power-down
  - in Deep Power-down

– during normal operation (clock stop)

Programming mode: Dynamic

Bit 2 – DEEPPOWERDOWN_EN
When this is 1, UDDRC puts the SDRAM into Deep Power-down mode when the transaction store is empty.
This register must be reset to '0' to bring UDDRC out of Deep Power-down mode. Controller performs automatic SDRAM initialization on deep power-down exit.
Present only in designs configured to support LPDDR2 or LPDDR3. For non-LPDDR2/non-LPDDR3, this register should not be set to 1.
For performance only.
Programming mode: Dynamic

Bit 1 – POWERDOWN_EN
If true then the UDDRC goes into Power-down after a programmable number of cycles "maximum idle clocks before power down" (PWRTMG.powerdown_to_x32).
This register bit may be re-programmed during the course of normal operation.
Programming mode: Dynamic

Bit 0 – SELFREF_EN
If true then the UDDRC puts the SDRAM into Self-refresh after a programmable number of cycles "maximum idle clocks before Self Refresh (PWRTMG.selfref_to_x32)". This register bit may be re-programmed during the course of normal operation.
Programming mode: Dynamic

17.6.10 UDDRC Low Power Timing Register

Name: UDDRC_PWRTMG
Offset: 0x034
Reset: 0x00402010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	SELFREF_TO_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	T_DPD_X4096[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				POWERDOWN_TO_X32[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	0	0	0	0

Bits 23:16 – SELFREF_TO_X32[7:0]
After this many clocks of the DDRC command channel being idle the UDDRC automatically puts the SDRAM into Self-refresh.
The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.selfref_en.
For performance only.
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Quasi-dynamic Group 4

Bits 15:8 – T_DPD_X4096[7:0] Minimum deep power-down time
For LPDDR2/LPDDR3, value from the JEDEC specification is 500 μs.
Present only in designs configured to support LPDDR2 or LPDDR3.
For performance only.
Unit: Multiples of 4096 DFI clock cycles.
Programming mode: Quasi-dynamic Group 4

Bits 4:0 – POWERDOWN_TO_X32[4:0]
After this many clocks of the DDRC command channel being idle the UDDRC automatically puts the SDRAM into Power-down. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.powerdown_en.
For performance only.
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Quasi-dynamic Group 4

17.6.11 UDDRC Hardware Low Power Control Register

Name: UDDRC_HWLPCTL
Offset: 0x038
Reset: 0x00000003
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					HW_LP_IDLE_X32[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HW_LP_IDLE_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							HW_LP_EXIT_IDLE_EN	HW_LP_EN
Access							R/W	R/W
Reset							1	1

Bits 27:16 – HW_LP_IDLE_X32[11:0] Hardware Idle Period
The cactive_ddrc output is driven low if the DDRC command channel is idle for hw_lp_idle * 32 cycles if not in Init or DPD/MPSM operating_mode. The DDRC command channel is considered idle when there are no HIF commands outstanding. The hardware idle function is disabled when hw_lp_idle_x32=0. hw_lp_idle_x32=1 is an illegal value.
For performance only.
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Static

Bit 1 – HW_LP_EXIT_IDLE_EN
When this bit is programmed to 1 the cactive_in_ddrc pin of the DDRC can be used to exit from the Automatic Clock Stop, Automatic Power-down or Automatic Self-refresh modes. Note, it will not cause exit of Self-refresh that was caused by hardware low-power Interface and/or software (PWRCTL.selfref_sw).
Programming mode: Static

Bit 0 – HW_LP_EN Enable for Hardware Low Power Interface
Programming mode: Quasi-dynamic Group 2

17.6.12 UDDRC Refresh Control Register 0

Name: UDDRC_RFSHCTL0
Offset: 0x050
Reset: 0x00210000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	REFRESH_MARGIN[3:0]							REFRESH_TO_X1_X32[4]
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	1	0				1

Bit	15	14	13	12	11	10	9	8
	REFRESH_TO_X1_X32[3:0]						REFRESH_BURST[5:4]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	REFRESH_BURST[3:0]					PER_BANK_REFRESH		
Access	R/W	R/W	R/W	R/W		R/W		
Reset	0	0	0	0		0		

Bits 23:20 – REFRESH_MARGIN[3:0]
Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom/32.
Note that internally used t_rfc_nom is equal to RFSHTMG.t_rfc_nom_x1_x32 * 32 if RFSHTMG.t_rfc_nom_x1_sel=0. If RFSHTMG.t_rfc_nom_x1_sel=1 (for LPDDR2/LPDDR3 per-bank refresh only), internally used t_rfc_nom is equal to RFSHTMG.t_rfc_nom_x1_x32. Note that, in LPDDR2/LPDDR3, internally used t_rfc_nom may be divided by four if derating is enabled (DERATEEN.derate_enable=1).
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Dynamic - Refresh Related

Bits 16:12 – REFRESH_TO_X1_X32[4:0]
If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x1_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continues successively until there are no refreshes pending or until new reads or writes are issued to the UDDRC.
For performance only.
Unit: DFI clock cycles or multiples of 32 DFI clock cycles, depending on RFSHTMG.t_rfc_nom_x1_sel.
Programming mode: Dynamic - Refresh Related

Bits 9:4 – REFRESH_BURST[5:0]
The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly increases utilization; lower numbers decreases the worst-case latency associated with refreshes.
For information on burst refresh feature refer to section 3.9 of DDR2 JEDEC specification - JESD79-2F.pdf.

For DDR2/3, the refresh is always per-rank and not per-bank. The rank refresh can be accumulated over 8*tREFI cycles using the burst refresh feature.
In per-bank refresh mode of LPDDR2/LPDDR3 (RFSHCTL0.per_bank_refresh = 1), 64 refreshes can be postponed.
Programming mode: Dynamic - Refresh Related

Value	Description
0	Single refresh
1	Burst-of-2 refresh
7	Burst-of-8 refresh

Bit 2 – PER_BANK_REFRESH
Per-bank refresh allows traffic to flow to other banks. Per-bank refresh is not supported by all LPDDR2 devices but should be supported by all LPDDR3 devices. Present only in designs configured to support LPDDR2/LPDDR3.
Programming mode: Static

Value	Description
0	All-bank refresh
1	Per-bank refresh

17.6.13 UDDRC Refresh Control Register 3

Name: UDDRC_RFSHCTL3
Offset: 0x060
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							REFRESH_UPDATE_LEVEL	DIS_AUTO_REFRESH
Access							R/W	R/W
Reset							0	0

Bit 1 – REFRESH_UPDATE_LEVEL
Toggle this signal (either from 0 to 1 or from 1 to 0) to indicate that the refresh register(s) have been updated.
refresh_update_level must not be toggled when the DDRC is in reset (core_ddrc_rstn = 0).
The refresh register(s) are automatically updated when exiting reset.
Programming mode: Dynamic

Bit 0 – DIS_AUTO_REFRESH
When '1', disable Auto-refresh generated by the UDDRC. When Auto-refresh is disabled, the SoC core must generate refreshes using the registers DBGCMD.rankn_refresh.
When dis_auto_refresh transitions from 0 to 1, any pending refreshes are immediately scheduled by the UDDRC.
This register field is changeable on the fly.
Programming mode: Dynamic - Refresh Related

17.6.14 UDDRC Refresh Timing Register

Name: UDDRC_RFSHTMG
Offset: 0x064
Reset: 0x0062008C
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	T_RFC_NOM_X1_SEL				T_RFC_NOM_X1_X32[11:8]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	T_RFC_NOM_X1_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	LPDDR3_TREFBW_EN						T_RFC_MIN[9:8]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	T_RFC_MIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	1	0	0

Bit 31 – T_RFC_NOM_X1_SEL
Specifies whether the RFSHTMG.t_rfc_nom_x1_x32 and RFSHCTL0.refresh_to_x1_x32 register values are x1 or x32.
This applies only when per-bank refresh is enabled (RFSHCTL0.per_bank_refresh=1); if per-bank refresh is not enabled, the x32 register values are used and this register field is ignored.
Programming mode: Dynamic - Refresh Related

Value	Description
0	x32 register values are used.
1	x1 register values are used.

Bits 27:16 – T_RFC_NOM_X1_X32[11:0]
Average time interval between refreshes per rank (Specification: 7.8 μs for DDR2 and DDR3. See JEDEC specification for LPDDR2 and LPDDR3).
When the controller is operating in 1:1 mode, set this register to RoundDown(tREFI/tCK).
When the controller is operating in 1:2 mode, set this register to RoundDown(RoundDown(tREFI/tCK)/2).
In both the above cases, if RFSHTMG.t_rfc_nom_x1_sel = 0, divide the above result by 32 and round down.
For LPDDR2/LPDDR3:

- If using all-bank refreshes (RFSHCTL0.per_bank_refresh = 0), use tREFIab in the above calculations.
- If using per-bank refreshes (RFSHCTL0.per_bank_refresh = 1), use tREFIpb in the above calculations.

Note that:

- RFSHTMG.t_rfc_nom_x1_x32 must be greater than 0x1.
- If RFSHTMG.t_rfc_nom_x1_sel == 1, RFSHTMG.t_rfc_nom_x1_x32 must be greater than RFSHTMG.t_rfc_min.
- If RFSHTMG.t_rfc_nom_x1_sel == 0, RFSHTMG.t_rfc_nom_x1_x32 * 32 must be greater than RFSHTMG.t_rfc_min.
- In Fixed 1x mode: RFSHTMG.t_rfc_nom_x1_x32 must be less than or equal to 0xFFE.

Unit: DFI clock cycles or multiples of 32 DFI clock cycles, depending on RFSHTMG.t_rfc_nom_x1_sel.

Programming mode: Dynamic - Refresh Related

Bit 15 – LPDDR3_TREFBW_EN
Used only when LPDDR3 memory type is connected. Should only be changed when UDDRC is in reset. Specifies whether to use the tREFBW parameter (required by some LPDDR3 devices which comply with earlier versions of the LPDDR3 JEDEC specification) or not.
Programming mode: Static

Value	Description
0	tREFBW parameter not used.
1	tREFBW parameter used.

Bits 9:0 – T_RFC_MIN[9:0] tRFC (min): Minimum time from refresh to refresh or activate.
When the controller is operating in 1:1 mode, t_rfc_min should be set to RoundUp(tRFCmin/tCK).
When the controller is operating in 1:2 mode, t_rfc_min should be set to RoundUp(RoundUp(tRFCmin/tCK)/2).
In LPDDR2/LPDDR3 mode:

- If using all-bank refreshes, the tRFCmin value in the above equations is equal to tRFCab.
- If using per-bank refreshes, the tRFCmin value in the above equations is equal to tRFCpb.

Unit: DFI clock cycles.
Programming mode: Dynamic - Refresh Related

17.6.15 UDDRC CRC Parity Control Register 0

Name: UDDRC_CRCPARCTL0
Offset: 0x0C0
Reset: 0x00000000
Property: Read/Write

Note: Do not perform any APB access to CRCPARCTL0 within 32 pclk cycles of previous access to CRCPARCTL0, as this might lead to data loss.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DFI_ALERT_ERR_CNT_CLR	DFI_ALERT_ERR_INT_CLR	DFI_ALERT_ERR_INT_EN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – DFI_ALERT_ERR_CNT_CLR DFI Alert Error Count Clear
Clear bit for DFI alert error counter. Asserting this bit will clear the DFI alert error counter, CRCPARSTAT.dfi_alert_err_cnt. UDDRC automatically clears this bit.
Programming mode: Dynamic

Bit 1 – DFI_ALERT_ERR_INT_CLR Interrupt Clear Bit for DFI Alert Error
If this bit is set, the alert error interrupt on CRCPARSTAT.dfi_alert_err_int will be cleared. UDDRC automatically clears this bit.
Programming mode: Dynamic

Bit 0 – DFI_ALERT_ERR_INT_EN Interrupt enable bit for DFI alert error
If this bit is set, any parity/CRC error detected on the dfi_alert_n input will result in an interrupt being set on CRCPARSTAT.dfi_alert_err_int.
Programming mode: Dynamic

17.6.16 UDDRC CRC Parity Status Register

Name: UDDRC_CRCPARSTAT
Offset: 0x0CC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								DFI_ALERT_ERR_INT
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
	DFI_ALERT_ERR_CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DFI_ALERT_ERR_CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 16 – DFI_ALERT_ERR_INT DFI Alert Error Interrupt
If a parity/CRC error is detected on dfi_alert_n, and the interrupt is enabled by CRCPARCTL0.dfi_alert_err_int_en, this interrupt bit will be set. It will remain set until cleared by CRCPARCTL0.dfi_alert_err_int_clr.
Programming Mode: Static

Bits 15:0 – DFI_ALERT_ERR_CNT[15:0] DFI Alert Error Count
If a parity/CRC error is detected on dfi_alert_n, this counter be incremented. This is independent of the setting of CRCPARCTL0.dfi_alert_err_int_en. It will saturate at 0xFFFF, and can be cleared by asserting CRCPARCTL0.dfi_alert_err_cnt_clr.
Programming Mode: Static

17.6.17 UDDRC SDRAM Initialization Register 0

Name: UDDRC_INIT0
Offset: 0x0D0
Reset: 0x0002004E
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SKIP_DRAM_INIT[1:0]						POST_CKE_X1024[9:8]	
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	23	22	21	20	19	18	17	16
	POST_CKE_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
					PRE_CKE_X1024[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PRE_CKE_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	1	1	0

Bits 31:30 – SKIP_DRAM_INIT[1:0]
If lower bit is enabled the SDRAM initialization routine is skipped. The upper bit decides what state the controller starts up in when reset is removed.
Programming mode: Quasi-dynamic Group 2

Value	Description
00	SDRAM Initialization routine is run after power-up.
01	SDRAM Initialization routine is skipped after power-up. Controller starts up in Normal mode.
11	SDRAM Initialization routine is skipped after power-up. Controller starts up in Self-refresh mode.
10	Reserved

Bits 25:16 – POST_CKE_X1024[9:0]
Cycles to wait after driving CKE high to start the SDRAM initialization sequence.
DDR2 typically requires a 400 ns delay, requiring this value to be programmed to 2 at all clock speeds.
LPDDR2/LPDDR3 typically requires this to be programmed for a delay of 200 µs.
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
Unit: Multiples of 1024 DFI clock cycles.
Programming mode: Static

Bits 11:0 – PRE_CKE_X1024[11:0]
Cycles to wait after reset before driving CKE high to start the SDRAM initialization sequence.
DDR2 specifications typically require this to be programmed for a delay of ≥200 µs.
LPDDR2/LPDDR3: tINIT1 of 100 ns (min)
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
For DDR3 RDIMMs, this should include the time needed to satisfy tSTAB.
Unit: Multiples of 1024 DFI clock cycles.
Programming mode: Static

17.6.18 UDDRC SDRAM Initialization Register 1

Name: UDDRC_INIT1
Offset: 0x0D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								DRAM_RSTN_X1024[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	DRAM_RSTN_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PRE_OCD_X32[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 24:16 – DRAM_RSTN_X1024[8:0]
Number of cycles to assert SDRAM reset signal during init sequence.
This is only present for designs supporting DDR3 devices. For use with a DDR PHY, this should be set to a minimum of 1.
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
Unit: Multiples of 1024 DFI clock cycles.
Programming mode: Static

Bits 3:0 – PRE_OCD_X32[3:0]
Wait period before driving the OCD complete command to SDRAM.
There is no known specific requirement for this; it may be set to zero.
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Static

17.6.19 UDDRC SDRAM Initialization Register 2

Name: UDDRC_INIT2
Offset: 0x0D8
Reset: 0x00000D05
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	IDLE_AFTER_RESET_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	1

Bit	7	6	5	4	3	2	1	0
					MIN_STABLE_CLOCK_X1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	1

Bits 15:8 – IDLE_AFTER_RESET_X32[7:0] Idle time after the reset command, tINIT4
Present only in designs configured to support LPDDR2.
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
Unit: Multiples of 32 DFI clock cycles.
Programming mode: Static

Bits 3:0 – MIN_STABLE_CLOCK_X1[3:0] Time to wait after the first CKE high, tINIT2
Present only in designs configured to support LPDDR2/LPDDR3.
LPDDR2/LPDDR3 typically requires 5 x tCK delay.
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
Unit: DFI clock cycles.
Programming mode: Static

17.6.20 UDDRC SDRAM Initialization Register 3

Name: UDDRC_INIT3
Offset: 0x0DC
Reset: 0x00000510
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EMR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	1

Bit	7	6	5	4	3	2	1	0
	EMR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

Bits 31:16 – MR[15:0]
DDR2: value to write to MR register. Bit 8 is for DLL and the setting here is ignored. The UDDRC sets this bit appropriately.
DDR3: value loaded into MR0 register.
LPDDR2/LPDDR3: value to write to MR1 register
Programming mode: Quasi-dynamic Group 1, Group 4

Bits 15:0 – EMR[15:0]
DDR2: Value to write to EMR register. Bits 9:7 are for OCD and the setting in this register is ignored. The UDDRC sets those bits appropriately.
DDR3: value to write to MR1 register. Set bit 7 to 0.
LPDDR2/LPDDR3: value to write to MR2 register
Programming mode: Quasi-dynamic Group 4

17.6.21 UDDRC SDRAM Initialization Register 4

Name: UDDRC_INIT4
Offset: 0x0E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EMR2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EMR2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EMR3[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EMR3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – EMR2[15:0] DDR2: Value to write to EMR2 register.
DDR3: Value to write to MR2 register
LPDDR2/LPDDR3: Value to write to MR3 register
Programming Mode: Quasi-dynamic Group 4

Bits 15:0 – EMR3[15:0] DDR2: Value to write to EMR3 register.
DDR3: Value to write to MR3 register
LPDDR2/LPDDR3: Unused
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.22 UDDRC SDRAM Initialization Register 5

Name: UDDRC_INIT5
Offset: 0x0E4
Reset: 0x00100004
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DEV_ZQINIT_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							MAX_AUTO_INIT_X1024[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	MAX_AUTO_INIT_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bits 23:16 – DEV_ZQINIT_X32[7:0] ZQ initial calibration, tZQINIT. Present only in designs configured to support DDR3 or LPDDR2/LPDDR3.
DDR3 typically requires 512 SDRAM clock cycles.
LPDDR2/LPDDR3 requires 1 µs.
When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.
Unit: Multiples of 32 DFI clock cycles.
Programming Mode: Static

Bits 9:0 – MAX_AUTO_INIT_X1024[9:0] Maximum duration of the auto initialization, tINIT5. Present only in designs configured to support LPDDR2/LPDDR3.
LPDDR2/LPDDR3 typically requires 10 µs.
Unit: Multiples of 1024 DFI clock cycles.
Programming Mode: Static

17.6.23 UDDRC DIMM Control Register

Name: UDDRC_DIMMCTL
Offset: 0x0F0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DIMM_ADDR_MIRR_EN	DIMM_STAGGER_CS_EN
Access							R/W	R/W
Reset							0	0

Bit 1 – DIMM_ADDR_MIRR_EN Address Mirroring Enable (for multi-rank UDIMM implementations).
Some UDIMMs implement address mirroring for odd ranks, which means that the following address, bank address and bank group bits are swapped: (A3, A4), (A5, A6), (A7, A8), (BA0, BA1). Setting this bit ensures that, for mode register accesses during the automatic initialization routine, these bits are swapped within the UDDRC to compensate for this UDIMM/RDIMM/LRDIMM swapping.
This is not supported for LPDDR2 or LPDDR3 SDRAMs.
Programming Mode: Static
Note: This has no effect on the address of any other memory accesses, or of software-driven mode register accesses.

Value	Description
0	For odd ranks, implement address mirroring for MRS commands to during initialization
1	Do not implement address mirroring

Bit 0 – DIMM_STAGGER_CS_EN Staggering enable for multi-rank accesses (for multi-rank UDIMM, RDIMM and LRDIMM implementations only). This is not supported for LPDDR2 or LPDDR3 SDRAMs.
Programming Mode: Static
Note: Even if this bit is set it does not take care of software driven MR commands (via MRCTRL0/MRCTRL1), where software is responsible to send them to separate ranks as appropriate.

Value	Description
0	Send all commands to even and odd ranks separately
1	Do not stagger accesses

17.6.24 UDDRC SDRAM Timing Register 0

Name: UDDRC_DRAMTMG0
Offset: 0x100
Reset: 0x0F101B0F
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		WR2PRE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	1	1	1	1

Bit	23	22	21	20	19	18	17	16
			T_FAW[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		T_RAS_MAX[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	1	1	0	1	1

Bit	7	6	5	4	3	2	1	0
			T_RAS_MIN[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	1	1	1	1

Bits 30:24 – WR2PRE[6:0] Minimum time between write and precharge to same bank.
Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400MHz and less for lower frequencies
where:
- WL = write latency
- BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present.
- tWR = Write recovery time. This comes directly from the SDRAM specification.

Add one extra cycle for LPDDR2/LPDDR3 for this parameter.
When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the above value by 2. No rounding up.
When the controller is operating in 1:2 frequency ratio mode or 2T mode, divide the above value by 2 and round it up to the next integer value.
Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 21:16 – T_FAW[5:0] tFAW: Valid only when 8 or more banks (or banks x bank groups) are present.
In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles.
When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value.
In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 14:8 – T_RAS_MAX[6:0] tRAS(max): Maximum time between activate and precharge to same bank. This is the maximum time that a page can be kept open
Minimum value of this register is 1. Zero is invalid.
When the controller is operating in 1:1 frequency ratio mode, t_ras_max should be set to RoundDown(tRAS(max)/tCK/1024).

When the controller is operating in 1:2 frequency ratio mode, t_ras_max should be set to RoundDown((RoundDown(tRAS(max)/tCK/1024)-1)/2).
Unit: Multiples of 1024 DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 5:0 – T_RAS_MIN[5:0] tRAS(min): Minimum time between activate and precharge to the same bank.
When the controller is operating in 1:2 frequency mode, 1T mode, program this to tRAS(min)/2. No rounding up.
When the controller is operating in 1:2 frequency ratio mode or 2T mode, program this to (tRAS(min)/2) and round it up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.25 UDDRC SDRAM Timing Register 1

Name: UDDRC_DRAMTMG1
Offset: 0x104
Reset: 0x00080414
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				T_XP[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
			RD2PRE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
		T_RC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	1	0	1	0	0

Bits 20:16 – T_XP[4:0] tXP: Minimum time after power-down exit to any operation. For DDR3, this should be programmed to tXPDLL if slow powerdown exit is selected in MR0[12].
When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value.
Units: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 13:8 – RD2PRE[5:0] tRTP: Minimum time from read to precharge of same bank.
- DDR2: tAL + BL/2 + max(tRTP, 2) - 2
- DDR3: tAL + max (tRTP, 4)
- LPDDR2: Depends on if it's LPDDR2-S2 or LPDDR2-S4:
LPDDR2-S2: BL/2 + tRTP - 1.
LPDDR2-S4: BL/2 + max(tRTP,2) - 2.
- LPDDR3: BL/2 + max(tRTP,4) - 4
When the controller is operating in 1:2 mode, 1T mode, divide the above value by 2. No rounding up.
When the controller is operating in 1:2 mode or 2T mode, divide the above value by 2 and round it up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 6:0 – T_RC[6:0] tRC: Minimum time between activates to same bank.
When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.26 UDDRC SDRAM Timing Register 2

Name: UDDRC_DRAMTMG2
Offset: 0x108
Reset: 0x0305060D
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			WRITE_LATENCY[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	1	1

Bit	23	22	21	20	19	18	17	16
			READ_LATENCY[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8
			RD2WR[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	1	0

Bit	7	6	5	4	3	2	1	0
			WR2RD[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	1	1	0	1

Bits 29:24 – WRITE_LATENCY[5:0] Set to WL
Time from write command to write data on SDRAM interface. This must be set to WL.
Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of WL to compensate for the extra cycle of latency through the RDIMM/LRDIMM.
When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.
This register field is not required for DDR2 and DDR3, as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols
For all protocols, in addition to programming this register field, it is necessary to program DFITMG0 and DFITMG1 to control the read and write latencies
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 21:16 – READ_LATENCY[5:0] Set to RL
Time from read command to read data on SDRAM interface. This must be set to RL.
Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of RL to compensate for the extra cycle of latency through the RDIMM/LRDIMM.
When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.
This register field is not required for DDR2 and DDR3, as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols
For all protocols, in addition to programming this register field, it is necessary to program DFITMG0 and DFITMG1 to control the read and write latencies
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 13:8 – RD2WR[5:0]
DDR2/3: RL + BL/2 + 2 - WL
LPDDR2/LPDDR3: RL + BL/2 + RU(tDQSCKmax/tCK) + 1 - WL

Minimum time from read command to write command. Include time for bus turnaround and all per-bank, per-rank, and global constraints.
Please see the relevant PHY databook for details of what should be included here.
Where:
- WL = write latency
- BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM
- RL = read latency = CAS latency
After PHY has completed training the value programmed may need to be increased. Refer to relevant PHY documentation.
For LPDDR2/LPDDR3, if derating is enabled (DERATEEN.derate_enable=1), derated tDQSCKmax should be used.
When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.
Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 5:0 – WR2RD[5:0]
LPDDR2/3: WL + BL/2 + tWTR + 1
Others: CWL + BL/2 + tWTR
Please see the relevant PHY databook for details of what should be included here.
Where:
- CWL = CAS write latency
- WL = Write latency
- PL = Parity latency
- BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM
- tWTR_L = internal write to read command delay for same bank group. This comes directly from the SDRAM specification.
- tWTR = internal write to read command delay. This comes directly from the SDRAM specification.
After PHY has completed training the value programmed may need to be increased. Refer to relevant PHY documentation.
Add one extra cycle for LPDDR2/LPDDR3 operation.
When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

17.6.27 UDDRC SDRAM Timing Register 3

Name: UDDRC_DRAMTMG3
Offset: 0x10C
Reset: 0x0050400C
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			T_MRW[9:4]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	T_MRW[3:0]						T_MRD[5:4]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	1	0	1			0	0

Bit	15	14	13	12	11	10	9	8
	T_MRD[3:0]						T_MOD[9:8]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	T_MOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

Bits 29:20 – T_MRW[9:0] Time to wait after a mode register write or read (MRW or MRR).
Present only in designs configured to support LPDDR2 or LPDDR3.
LPDDR2 typically requires value of 5.
LPDDR3 typically requires value of 10.
For LPDDR2, this register is used for the time from a MRW/MRR to all other commands.
When the controller is operating in 1:2 frequency ratio mode, program this to the above values divided by 2 and round it up to the next integer value.
For LDPDR3, this register is used for the time from a MRW/MRR to a MRW/MRR.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 17:12 – T_MRD[5:0] tMRD: Cycles to wait after a mode register write or read. Depending on the connected SDRAM, tMRD represents:
DDR2: Time from MRS to any command
DDR3: Time from MRS to MRS command
LPDDR2: not used
LPDDR3: Time from MRS to non-MRS command.
When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value.
If CAL mode is enabled (DFITMG1.dfi_t_cmd_lat > 0), tCAL (=DFITMG1.dfi_cmd_lat) should be added to the above calculations.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 9:0 – T_MOD[9:0] tMOD: Parameter used only in DDR3. Cycles between load mode command and following non-load mode command.
If CAL mode is enabled (DFITMG1.dfi_t_cmd_lat > 0), tCAL (=DFITMG1.dfi_cmd_lat) should be added to the above calculations.
Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode. Note that if using RDIMM/LRDIMM, depending on the PHY, it may be

necessary to adjust the value of this parameter to compensate for the extra cycle of latency applied to mode register writes by the RDIMM/LRDIMM chip.
Also note that if using LRDIMM, the minimum value of this register is tMRD_L2 if controller is operating in 1:1 frequency ratio mode, or tMRD_L2/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.28 UDDRC SDRAM Timing Register 4

Name: UDDRC_DRAMTMG4
Offset: 0x110
Reset: 0x05040405
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				T_RCD[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	0	1

Bit	23	22	21	20	19	18	17	16
					T_CCD[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	15	14	13	12	11	10	9	8
					T_RRD[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	7	6	5	4	3	2	1	0
				T_RP[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	0	1

Bits 28:24 – T_RCD[4:0] tRCD - tAL: Minimum time from activate to read or write command to same bank.
When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value.
Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 2, Group 4

Bits 19:16 – T_CCD[3:0] tCCD: This is the minimum time between two reads or two writes
When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 11:8 – T_RRD[3:0] tRRD: Minimum time between activates from bank "a" to bank "b"
When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_L/2 or tRRD/2) and round it up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 4:0 – T_RP[4:0] tRP: Minimum time from single-bank precharge to activate of same bank.
When the controller is operating in 1:1 frequency ratio mode, t_rp should be set to RoundUp(tRP/tCK).
When the controller is operating in 1:2 frequency ratio mode, t_rp should be set to RoundDown(RoundUp(tRP/tCK)/2) + 1.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.29 UDDRC SDRAM Timing Register 5

Name: UDDRC_DRAMTMG5
Offset: 0x114
Reset: 0x05050403
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					T_CKSRX[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	1

Bit	23	22	21	20	19	18	17	16
					T_CKSRE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	1

Bit	15	14	13	12	11	10	9	8
			T_CKESR[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
				T_CKE[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	1	1

Bits 27:24 – T_CKSRX[3:0] This is the time before Self Refresh Exit that CK is maintained as a valid clock before issuing SRX. Specifies the clock stable time before SRX.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2
- DDR2: 1
- DDR3: tCKSRX
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 19:16 – T_CKSRE[3:0] This is the time after Self Refresh Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after SRE.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2
- DDR2: 1
- DDR3: max (10 ns, 5 tCK)
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 13:8 – T_CKESR[5:0] Minimum CKE low width for Self refresh or Self refresh power down entry to exit timing in memory clock cycles.
Recommended settings:
- LPDDR2: tCKESR
- LPDDR3: tCKESR
- DDR2: tCKE
- DDR3: tCKE + 1
Unit: DFI clock cycles.

Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 4:0 – T_CKE[4:0] Minimum number of cycles of CKE HIGH/LOW during power-down and self refresh.
- LPDDR2/LPDDR3 mode: Set this to the larger of tCKE or tCKESR
- Non-LPDDR2/non-LPDDR3 designs: Set this to tCKE value.
When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.30 UDDRC SDRAM Timing Register 6

Name: UDDRC_DRAMTMG6
Offset: 0x118
Reset: 0x02020005
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					T_CKDPDE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit	23	22	21	20	19	18	17	16
					T_CKDPDX[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					T_CKCSX[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	1

Bits 27:24 – T_CKDPDE[3:0] This is the time after Deep Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after DPDE.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 19:16 – T_CKDPDX[3:0] This is the time before Deep Power Down Exit that CK is maintained as a valid clock before issuing DPDX. Specifies the clock stable time before DPDX.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
This is only present for designs supporting LPDDR2 devices.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 3:0 – T_CKCSX[3:0] This is the time before Clock Stop Exit that CK is maintained as a valid clock before issuing Clock Stop Exit. Specifies the clock stable time before next command after Clock Stop Exit.
Recommended settings:
- LPDDR2: tXP + 2
- LPDDR3: tXP + 2
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
This is only present for designs supporting LPDDR2/LPDDR3 devices.

Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.31 UDDRC SDRAM Timing Register 7

Name: UDDRC_DRAMTMG7
Offset: 0x11C
Reset: 0x00000202
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					T_CKPDE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit	7	6	5	4	3	2	1	0
					T_CKPDX[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bits 11:8 – T_CKPDE[3:0] This is the time after Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after PDE.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2

When using DDR2/3 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksre.
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

Bits 3:0 – T_CKPDX[3:0] This is the time before Power Down Exit that CK is maintained as a valid clock before issuing PDX. Specifies the clock stable time before PDX.
Recommended settings:
- LPDDR2: 2
- LPDDR3: 2

When using DDR2/3 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksrx.
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.32 UDDRC SDRAM Timing Register 8

Name: UDDRC_DRAMTMG8
Offset: 0x120
Reset: 0x00004405
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		T_XS_DLL_X32[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
		T_XS_X32[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	1	0	1

Bits 14:8 – T_XS_DLL_X32[6:0] tXSDLL: Exit Self Refresh to commands requiring a locked DLL.
When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value.
Unit: Multiples of 32 DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4
Note: Used only for DDR2 and DDR3 SDRAMs.

Bits 6:0 – T_XS_X32[6:0] tXS: Exit Self Refresh to commands not requiring a locked DLL.
When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value.
Unit: Multiples of 32 DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4
Note: Used only for DDR2 and DDR3 SDRAMs.

17.6.33 UDDRC SDRAM Timing Register 14

Name: UDDRC_DRAMTMG14
Offset: 0x138
Reset: 0x000000A0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					T_XSR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	T_XSR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	0	0	0	0

Bits 11:0 – T_XSR[11:0] tXSR: Exit Self Refresh to any command.
When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value.
The value 0xfff is illegal for this register field.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4
Note: Used only for LPDDR2/LPDDR3 mode.

17.6.34 UDDRC SDRAM Timing Register 15

Name: UDDRC_DRAMTMG15
Offset: 0x13C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EN_DFI_LP_T_STAB							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	T_STAB_X32[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – EN_DFI_LP_T_STAB
Programming Mode: Quasi-dynamic Group 2, Group 4

Value	Description
0	Disable using tSTAB when exiting DFI LP.
1	Enable using tSTAB when exiting DFI LP. Needs to be set when the PHY is stopping the clock during DFI LP to save maximum power.

Bits 7:0 – T_STAB_X32[7:0] tSTAB: Stabilization time.
It is required in the following two cases for DDR3 RDIMM:

- when exiting power saving mode, if the clock was stopped, after re-enabling it the clock must be stable for a time specified by tSTAB
- after issuing control words that refers to clock timing

(Specification: 6 μs for DDR3)
When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
Unit: Multiples of 32 DFI clock cycles.
Programming Mode: Quasi-dynamic Group 2, Group 4

17.6.35 UDDRC ZQ Control Register 0

Name: UDDRC_ZQCTL0
Offset: 0x180
Reset: 0x02000040
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DIS_AUTO_ZQ	DIS_SRX_ZQCL	ZQ_RESISTOR_SHARED			T_ZQ_LONG_NOP[10:8]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	1	0

Bit	23	22	21	20	19	18	17	16
	T_ZQ_LONG_NOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							T_ZQ_SHORT_NOP[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	T_ZQ_SHORT_NOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit 31 – DIS_AUTO_ZQ
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Programming Mode: Dynamic

Value	Description
0	Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024.
1	Disable UDDRC generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module.

Bit 30 – DIS_SRX_ZQCL
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Programming Mode: Quasi-dynamic Group 2, Group 4

Value	Description
0	Enable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3, LPDDR2 or LPDDR3 mode.
1	Disable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3, LPDDR2 or LPDDR3 mode.

Bit 29 – ZQ_RESISTOR_SHARED
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Programming Mode: Static

Value	Description
0	ZQ resistor is not shared.
1	Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap.

Bits 26:16 – T_ZQ_LONG_NOP[10:0] tZQoper for DDR3, tZQCL for LPDDR2/LPDDR3: Number of DFI clock cycles of NOP required after a ZQCL (ZQ calibration long)/MPC(ZQ Start) command is issued to SDRAM.
When the controller is operating in 1:2 frequency ratio mode:
DDR3: program this to tZQoper/2 and round it up to the next integer value.
LPDDR2/LPDDR3: program this to tZQCL/2 and round it up to the next integer value.
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Static

Bits 9:0 – T_ZQ_SHORT_NOP[9:0] tZQCS for DDR3/DD4/LPDDR2/LPDDR3: Number of DFI clock cycles of NOP required after a ZQCS (ZQ calibration short)/MPC(ZQ Latch) command is issued to SDRAM.
When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value.
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Static

17.6.36 UDDRC ZQ Control Register 1

Name: UDDRC_ZQCTL1
Offset: 0x184
Reset: 0x02000100
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			T_ZQ_RESET_NOP[9:4]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	1	0

Bit	23	22	21	20	19	18	17	16
	T_ZQ_RESET_NOP[3:0]				T_ZQ_SHORT_INTERVAL_X1024[19:16]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	T_ZQ_SHORT_INTERVAL_X1024[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	T_ZQ_SHORT_INTERVAL_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:20 – T_ZQ_RESET_NOP[9:0] tZQReset: Number of DFI clock cycles of NOP required after a ZQReset (ZQ calibration Reset) command is issued to SDRAM.
When the controller is operating in 1:2 frequency ratio mode, program this to tZQReset/2 and round it up to the next integer value.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.
Programming Mode: Static

Bits 19:0 – T_ZQ_SHORT_INTERVAL_X1024[19:0] Average interval to wait between automatically issuing ZQCS (ZQ calibration short)/MPC(ZQ calibration) commands to DDR3/LPDDR2/LPDDR3 devices.
Meaningless, if ZQCTL0.dis_auto_zq=1.
This is only present for designs supporting DDR3 or LPDDR2/LPDDR3 devices.
Unit: Multiples of 1024 DFI clock cycles.
Programming Mode: Static

17.6.37 UDDRC ZQ Control Register 2

Name: UDDRC_ZQCTL2
Offset: 0x188
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ZQ_RESET
Access								R/W
Reset								0

Bit 0 – ZQ_RESET
Setting this register bit to 1 triggers a ZQ Reset operation. When the ZQ Reset operation is complete, the UDDRC automatically clears this bit. It is recommended NOT to set this register bit if in Init, in Self-Refresh or Deep power-down operating modes.
For Self-Refresh or SR-Powerdown it will be scheduled after SR has been exited.
For Deep power down, it will not be scheduled, although ZQSTAT.zq_reset_busy will be de-asserted.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Programming Mode: Dynamic

17.6.38 UDDRC ZQ Status Register

Name: UDDRC_ZQSTAT
Offset: 0x18C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ZQ_RESET_BUSY
Access								R
Reset								0

Bit 0 – ZQ_RESET_BUSY SoC core may initiate a ZQ Reset operation only if this signal is low. This signal goes high in the clock after the UDDRC accepts the ZQ Reset request. It goes low when the ZQ Reset command is issued to the SDRAM and the associated NOP period is over. It is recommended not to perform ZQ Reset commands when this signal is high.
Programming Mode: Dynamic

Value	Description
0	Indicates that the SoC core can initiate a ZQ Reset operation.
1	Indicates that ZQ Reset operation is in progress.

17.6.39 UDDRC DFI Timing Register 0

Name: UDDRC_DFITMG0
Offset: 0x190
Reset: 0x07020002
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				DFI_T_CTRL_DELAY[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	DFI_RDDATA_USE_DFI_PHY_CLK	DFI_T_RDDATA_EN[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	DFI_WRDATA_USE_DFI_PHY_CLK		DFI_TPHY_WRDATA[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			DFI_TPHY_WRLAT[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	1	0

Bits 28:24 – DFI_T_CTRL_DELAY[4:0] Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value. Note that if using RDIMM/LRDIMM, it is necessary to increment this parameter by RDIMM's/LRDIMM's extra cycle of latency in terms of DFI clock.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 4

Bit 23 – DFI_RDDATA_USE_DFI_PHY_CLK Defines whether dfi_rddata_en/dfi_rddata/dfi_rddata_valid is generated using HDR (DFI clock) or SDR (DFI PHY clock) values.
Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles:
- 0 in terms of HDR (DFI clock) cycles
- 1 in terms of SDR (DFI PHY clock) cycles
Refer to PHY specification for correct value.
Programming Mode: Static

Bits 22:16 – DFI_T_RDDATA_EN[6:0] Time from the assertion of a read command on the DFI interface to the assertion of the dfi_rddata_en signal.
Refer to PHY specification for correct value.
This corresponds to the DFI parameter trddata_en. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of trddata_en. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM.
Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_dfi_phy_clk.
Programming Mode: Quasi-dynamic Group 1, Group 4

Bit 15 – DFI_WRDATA_USE_DFI_PHY_CLK Defines whether dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is generated using HDR (DFI clock) or SDR (DFI PHY clock) values
Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles
Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles
- 0 in terms of HDR (DFI clock) cycles
- 1 in terms of SDR (DFI PHY clock) cycles
Refer to PHY specification for correct value.
Programming Mode: Static

Bits 13:8 – DFI_TPHY_WRDATA[5:0] Specifies the number of clock cycles between when dfi_wrdata_en is asserted to when the associated write data is driven on the dfi_wrdata signal. This corresponds to the DFI timing parameter tphy_wrdata. Refer to PHY specification for correct value. Note, max supported value is 8.
Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_dfi_phy_clk.
Programming Mode: Quasi-dynamic Group 4

Bits 5:0 – DFI_TPHY_WRLAT[5:0] Write latency
Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat.
Refer to PHY specification for correct value. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of tphy_wrlat. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM.
Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_dfi_phy_clk.
Programming Mode: Quasi-dynamic Group 1, Group 4

17.6.40 UDDRC DFI Timing Register 1

Name: UDDRC_DFITMG1
Offset: 0x194
Reset: 0x00000404
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							DFI_T_PARIN_LAT[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				DFI_T_WRDATA_DELAY[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFI_T_DRAM_CLK_DISABLE[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
				DFI_T_DRAM_CLK_ENABLE[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	0	0

Bits 25:24 – DFI_T_PARIN_LAT[1:0] Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated dfi_parity_in signal is driven.
Unit: DFI PHY clock cycles.
Programming Mode: Quasi-dynamic Group 4

Bits 20:16 – DFI_T_WRDATA_DELAY[4:0] Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted and when the corresponding write data transfer is completed on the DRAM bus.
This corresponds to the DFI timing parameter twrdata_delay. Refer to PHY specification for correct value.
For DFI 3.0 PHY, set to twrdata_delay, a new timing parameter introduced in DFI 3.0.
For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM).
Value to be programmed is in terms of DFI clocks, not PHY clocks.
In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer.
If using DFITMG0.dfi_wrdata_use_dfi_phy_clk=1, add 1 to the value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 4

Bits 12:8 – DFI_T_DRAM_CLK_DISABLE[4:0] Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 4

Bits 4:0 – DFI_T_DRAM_CLK_ENABLE[4:0] Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 4

17.6.41 UDDRC DFI Low Power Configuration Register 0

Name: UDDRC_DFILPCFG0
Offset: 0x198
Reset: 0x07000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				DFI_TLP_RESP[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	DFI_LP_WAKEUP_DPD[3:0]							DFI_LP_EN_DPD
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
	DFI_LP_WAKEUP_SR[3:0]							DFI_LP_EN_SR
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
	DFI_LP_WAKEUP_PD[3:0]							DFI_LP_EN_PD
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 28:24 – DFI_TLP_RESP[4:0] Setting in DFI clock cycles for DFI's tlp_resp time.
Same value is used for both Power Down, Self Refresh, Deep Power Down and Maximum Power Saving modes.
Refer to PHY databook for recommended values
Unit: DFI clock cycles.
Programming Mode: Static

Bits 23:20 – DFI_LP_WAKEUP_DPD[3:0] Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered.
Determines the DFI's tlp_wakeup time:
- 0x0 - 16 cycles
- 0x1 - 32 cycles
- 0x2 - 64 cycles
- 0x3 - 128 cycles
- 0x4 - 256 cycles
- 0x5 - 512 cycles
- 0x6 - 1024 cycles
- 0x7 - 2048 cycles
- 0x8 - 4096 cycles
- 0x9 - 8192 cycles
- 0xA - 16384 cycles
- 0xB - 32768 cycles
- 0xC - 65536 cycles
- 0xD - 131072 cycles
- 0xE - 262144 cycles
- 0xF - Unlimited

This is only present for designs supporting LPDDR2/LPDDR3 devices.
Unit: DFI clock cycles.

Programming Mode: Static

Bit 16 – DFI_LP_EN_DPD Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit.
This is only present for designs supporting LPDDR2/LPDDR3 devices.
Programming Mode: Static

Value	Description
0	Disabled
1	Enabled

Bits 15:12 – DFI_LP_WAKEUP_SR[3:0] Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Self Refresh mode is entered.
Determines the DFI's tlp_wakeup time:
- 0x0 - 16 cycles
- 0x1 - 32 cycles
- 0x2 - 64 cycles
- 0x3 - 128 cycles
- 0x4 - 256 cycles
- 0x5 - 512 cycles
- 0x6 - 1024 cycles
- 0x7 - 2048 cycles
- 0x8 - 4096 cycles
- 0x9 - 8192 cycles
- 0xA - 16384 cycles
- 0xB - 32768 cycles
- 0xC - 65536 cycles
- 0xD - 131072 cycles
- 0xE - 262144 cycles
- 0xF - Unlimited

Unit: DFI clock cycles.
Programming Mode: Static

Bit 8 – DFI_LP_EN_SR Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit.
Programming Mode: Static

Value	Description
0	Disabled
1	Enabled

Bits 7:4 – DFI_LP_WAKEUP_PD[3:0] Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Power Down mode is entered.
Determines the DFI's tlp_wakeup time:
- 0x0 - 16 cycles
- 0x1 - 32 cycles
- 0x2 - 64 cycles
- 0x3 - 128 cycles
- 0x4 - 256 cycles
- 0x5 - 512 cycles
- 0x6 - 1024 cycles
- 0x7 - 2048 cycles
- 0x8 - 4096 cycles
- 0x9 - 8192 cycles
- 0xA - 16384 cycles
- 0xB - 32768 cycles
- 0xC - 65536 cycles
- 0xD - 131072 cycles
- 0xE - 262144 cycles
- 0xF - Unlimited

Unit: DFI clock cycles.
Programming Mode: Static

Bit 0 – DFI_LP_EN_PD Enables DFI Low Power interface handshaking during Power Down Entry/Exit.
Programming Mode: Static

Value	Description
0	Disabled
1	Enabled

17.6.42 UDDRC DFI Update Register 0

Name: UDDRC_DFIUPD0
Offset: 0x1A0
Reset: 0x00400003
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DIS_AUTO_CTRLUPD	DIS_AUTO_CTRLUPD_SRX	CTRLUPD_PRE_SRX				DFI_T_CTRLUP_MAX[9:8]	
Access	R/W	R/W	R/W				R/W	R/W
Reset	0	0	0				0	0

Bit	23	22	21	20	19	18	17	16
	DFI_T_CTRLUP_MAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							DFI_T_CTRLUP_MIN[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	DFI_T_CTRLUP_MIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bit 31 – DIS_AUTO_CTRLUPD When '1', disable the automatic dfi_ctrlupd_req generation by the UDDRC. The core must issue the dfi_ctrlupd_req signal using register DBGCMD.ctrlupd.
When '0', UDDRC issues dfi_ctrlupd_req periodically.
Programming Mode: Quasi-dynamic Group 3

Bit 30 – DIS_AUTO_CTRLUPD_SRX When '1', disable the automatic dfi_ctrlupd_req generation by the UDDRC at self-refresh exit.
When '0', UDDRC issues a dfi_ctrlupd_req before or after exiting self-refresh, depending on DFIUPD0.ctrlupd_pre_srx.
Programming Mode: Static

Bit 29 – CTRLUPD_PRE_SRX Selects dfi_ctrlupd_req requirements at SRX:
If DFIUPD0.dis_auto_ctrlupd_srx=1, this register has no impact, because no dfi_ctrlupd_req will be issued when SRX.
Programming Mode: Static

Value	Description
0	Send ctrlupd after SRX.
1	Send ctrlupd before SRX.

Bits 25:16 – DFI_T_CTRLUP_MAX[9:0] Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert. Lowest value to assign to this variable is 0x40.
Unit: DFI clock cycles.
Programming Mode: Static

Bits 9:0 – DFI_T_CTRLUP_MIN[9:0] Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted. The UDDRC expects the PHY to respond within this time. If the PHY does not respond, the UDDRC will de-assert dfi_ctrlupd_req after dfi_t_ctrlup_min + 2 cycles. Lowest value to assign to this variable is 0x3.
Unit: DFI clock cycles.

Programming Mode: Static

17.6.43 UDDRC DFI Update Register 1

Name: UDDRC_DFIUPD1
Offset: 0x1A4
Reset: 0x00010001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 23:16 – DFI_T_CTRLUPD_INTERVAL_MIN_X1024[7:0] This is the minimum amount of time between UDDRC initiated DFI update requests (which is executed whenever the UDDRC is idle). Set this number higher to reduce the frequency of update requests, which can have a small impact on the latency of the first read request when the UDDRC is idle. Minimum allowed value for this field is 1.
Unit: Multiples of 1024 DFI clock cycles.
Programming Mode: Static

Bits 7:0 – DFI_T_CTRLUPD_INTERVAL_MAX_X1024[7:0] This is the maximum amount of time between UDDRC initiated DFI update requests. This timer resets with each update request; when the timer expires dfi_ctrlupd_req is sent and traffic is blocked until the dfi_ctrlupd_ackx is received. PHY can use this idle time to recalibrate the delay lines to the DLLs. The DFI controller update is also used to reset PHY FIFO pointers in case of data capture errors. Updates are required to maintain calibration over PVT, but frequent updates may impact performance. Minimum allowed value for this field is 1.
Unit: Multiples of 1024 DFI clock cycles.
Programming Mode: Static
Note: Value programmed for DFIUPD1.dfi_t_ctrlupd_interval_max_x1024 must be greater than DFIUPD1.dfi_t_ctrlupd_interval_min_x1024.

17.6.44 UDDRC DFI Update Register 2

Name: UDDRC_DFIUPD2
Offset: 0x1A8
Reset: 0x80000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DFI_PHYUPD_EN							
Access	R/W							
Reset	1							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – DFI_PHYUPD_EN Enables the support for acknowledging PHY-initiated updates.
Programming Mode: Static

Value	Description
0	Disabled
1	Enabled

17.6.45 UDDRC DFI Miscellaneous Control Register

Name: UDDRC_DFIMISC
Offset: 0x1B0
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DFI_FREQUENCY[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			DFI_INIT_START	CTL_IDLE_EN				DFI_INIT_COMPLETE_EN
Access			R/W	R/W				R/W
Reset			0	0				1

Bits 12:8 – DFI_FREQUENCY[4:0] Indicates the operating frequency of the system. The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY.
Programming Mode: Quasi-dynamic Group 1

Bit 5 – DFI_INIT_START PHY init start request signal. When asserted it triggers the PHY init start request.
Programming Mode: Quasi-dynamic Group 3

Bit 4 – CTL_IDLE_EN Enables support of ctl_idle signal.
Programming Mode: Static

Bit 0 – DFI_INIT_COMPLETE_EN PHY initialization complete enable signal.
When asserted the dfi_init_complete signal can be used to trigger SDRAM initialisation
Programming Mode: Quasi-dynamic Group 3

17.6.46 UDDRC DFI Status Register

Name: UDDRC_DFISTAT
Offset: 0x1BC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DFI_LP_ACK	DFI_INIT_COMPLETE
Access							R	R
Reset							0	0

Bit 1 – DFI_LP_ACK Stores the value of the dfi_lp_ack input to the controller.
Programming Mode: Dynamic

Bit 0 – DFI_INIT_COMPLETE The status flag register which announces when the DFI initialization has been completed. The DFI INIT triggered by dfi_init_start signal and then the dfi_init_complete flag is polled to know when the initialization is done.
Programming Mode: Dynamic

17.6.47 UDDRC DFI PHY Host

Name: UDDRC_DFIPHYMSTR
Offset: 0x1C4
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DFI_PHYMSTR_EN
Access								R/W
Reset								1

Bit 0 – DFI_PHYMSTR_EN Enables the PHY Host Interface
Programming Mode: Static

Value	Description
0	Disabled
1	Enabled

17.6.48 UDDRC Address Map Register 1

Name: UDDRC_ADDRMAP1
Offset: 0x204
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			ADDRMAP_BANK_B2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
			ADDRMAP_BANK_B1[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
			ADDRMAP_BANK_B0[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 21:16 – ADDRMAP_BANK_B2[5:0] Selects the HIF address bit used as bank address bit 2.
Valid Range: 0 to 31, and 63
Internal Base: 4
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 63 and then bank address bit 2 is set to 0.
Programming Mode: Static

Bits 13:8 – ADDRMAP_BANK_B1[5:0] Selects the HIF address bits used as bank address bit 1.
Valid Range: 0 to 32, and 63
Internal Base: 3
The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.
If unused, set to 63 and then bank address bit 1 is set to 0.
Programming Mode: Static

Bits 5:0 – ADDRMAP_BANK_B0[5:0] Selects the HIF address bits used as bank address bit 0.
Valid Range: 0 to 32, and 63
Internal Base: 2
The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.
If unused, set to 63 and then bank address bit 0 is set to 0.
Programming Mode: Static

17.6.49 UDDRC Address Map Register 2

Name: UDDRC_ADDRMAP2
Offset: 0x208
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					ADDRMAP_COL_B5[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
					ADDRMAP_COL_B4[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
				ADDRMAP_COL_B3[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					ADDRMAP_COL_B2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – ADDRMAP_COL_B5[3:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 5.
- Half bus width mode: Selects the HIF address bit used as column address bit 6.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 7.

Valid Range: 0 to 7, and 15
Internal Base: 5
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then this column address bit is set to 0.
Programming Mode: Static

Bits 19:16 – ADDRMAP_COL_B4[3:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 4.
- Half bus width mode: Selects the HIF address bit used as column address bit 5.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 6.

Valid Range: 0 to 7, and 15
Internal Base: 4
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then this column address bit is set to 0.
Programming Mode: Static

Bits 12:8 – ADDRMAP_COL_B3[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 3.
- Half bus width mode: Selects the HIF address bit used as column address bit 4.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 5.

Valid Range: 0 to 7.
Internal Base: 3
The selected HIF address bit is determined by adding the internal base to the value of this field.

Programming Mode: Static

Bits 3:0 – ADDRMAP_COL_B2[3:0]

- Full bus width mode: Selects the HIF address bit used as column address bit 2.
- Half bus width mode: Selects the HIF address bit used as column address bit 3.

Valid Range: 0 to 7
Internal Base: 2
The selected HIF address bit is determined by adding the internal base to the value of this field.
Note it is required to program this to 0 unless:

- in Half bus width (MSTR.data_bus_width!=00) and
- PCCFG.bl_exp_mode==1

Otherwise, if Full Bus Width (MSTR.data_bus_width)==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2.
Programming Mode: Static

17.6.50 UDDRC Address Map Register 3

Name: UDDRC_ADDRMAP3
Offset: 0x20C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				ADDRMAP_COL_B9[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				ADDRMAP_COL_B8[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				ADDRMAP_COL_B7[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				ADDRMAP_COL_B6[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 28:24 – ADDRMAP_COL_B9[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 9.
- Half bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode).
- Quarter bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode).
Valid Range: 0 to 7, and 31.
Internal Base: 9
The selected HIF address bit is determined by adding the internal base to the value of this field.
In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
If column bit 9 is the highest column address bit, it must map to the highest valid HIF address bit. (x = the highest valid HIF address bit - internal base)
If column bit 9 is the second highest column address bit, it must map to the second highest valid HIF address bit. (x = the highest valid HIF address bit - 1 - internal base)
If column bit 9 is the third highest column address bit, it must map to the third highest valid HIF address bit. (x = the highest valid HIF address bit - 2 - internal base)
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static
Note: Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10.

Bits 20:16 – ADDRMAP_COL_B8[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 8.
- Half bus width mode: Selects the HIF address bit used as column address bit 9.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode).
Valid Range: 0 to 7, and 31.
Internal Base: 8
The selected HIF address bit is determined by adding the internal base to the value of this field.
In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.

If column bit 8 is the second highest column address bit, it must map to the second highest valid HIF address bit. (x = the highest valid HIF address bit - 1 - internal base)
If column bit 8 is the third highest column address bit, it must map to the third highest valid HIF address bit. (x = the highest valid HIF address bit - 2 - internal base)
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static
Note: Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10.

Bits 12:8 – ADDRMAP_COL_B7[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 7.
- Half bus width mode: Selects the HIF address bit used as column address bit 8.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 9.
Valid Range: 0 to 7, and 31.
Internal Base: 7
The selected HIF address bit is determined by adding the internal base to the value of this field.
If column bit 7 is the third highest column address bit, it must map to the third highest valid HIF address bit. (x = the highest valid HIF address bit - 2 - internal base)
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static

Bits 4:0 – ADDRMAP_COL_B6[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 6.
- Half bus width mode: Selects the HIF address bit used as column address bit 7.
- Quarter bus width mode: Selects the HIF address bit used as column address bit 8.
Valid Range: 0 to 7.
Internal Base: 6
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static

17.6.51 UDDRC Address Map Register 4

Name: UDDRC_ADDRMAP4
Offset: 0x210
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				ADDRMAP_COL_B11[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				ADDRMAP_COL_B10[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – ADDRMAP_COL_B11[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode).
- Half bus width mode: UNUSED. See later in this description for value you need to set to make it unused.
- Quarter bus width mode: UNUSED. See later in this description for value you need to set to make it unused.
Valid Range: 0 to 7, and 31.
Internal Base: 11
The selected HIF address bit is determined by adding the internal base to the value of this field.
In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
If column bit 11 is the highest column address bit, it must map to the highest valid HIF address bit. (x = the highest valid HIF address bit - internal base)
If column bit 11 is the second highest column address bit, it must map to the second highest valid HIF address bit. (x = the highest valid HIF address bit - 1 - internal base)
If column bit 11 is the third highest column address bit, it must map to the third highest valid HIF address bit. (x = the highest valid HIF address bit - 2 - internal base)
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static
Note: Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10.

Bits 4:0 – ADDRMAP_COL_B10[4:0]
- Full bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode).
- Half bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode).
- Quarter bus width mode: UNUSED. See later in this description for value you need to set to make it unused.
Valid Range: 0 to 7, and 31.
Internal Base: 10
The selected HIF address bit is determined by adding the internal base to the value of this field.
In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
If column bit 10 is the highest column address bit, it must map to the highest valid HIF address bit. (x = the highest valid HIF address bit - internal base)

If column bit 10 is the second highest column address bit, it must map to the second highest valid HIF address bit. (x = the highest valid HIF address bit - 1 - internal base)
If column bit 10 is the third highest column address bit, it must map to the third highest valid HIF address bit. (x = the highest valid HIF address bit - 2 - internal base)
If unused, set to 31 and then this column address bit is set to 0.
Programming Mode: Static
Note: Per JEDEC DDR2/3 specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10.

17.6.52 UDDRC Address Map Register 5

Name: UDDRC_ADDRMAP5
Offset: 0x214
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					ADDRMAP_ROW_B11[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					ADDRMAP_ROW_B2_10[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					ADDRMAP_ROW_B1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					ADDRMAP_ROW_B0[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – ADDRMAP_ROW_B11[3:0] Selects the HIF address bit used as row address bit 11.
Valid Range: 0 to 11, and 15
Internal Base: 17
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then row address bit 11 is set to 0.
Programming Mode: Static

Bits 19:16 – ADDRMAP_ROW_B2_10[3:0] Selects the HIF address bits used as row address bits 2 to 10.
Valid Range: 0 to 11, and 15
Internal Base: 8 (for row address bit 2), 9 (for row address bit 3), 10 (for row address bit 4) etc increasing to 16 (for row address bit 10)
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.
When set to 15, the values of row address bits 2 to 10 are defined by registers ADDRMAP9, ADDRMAP10, ADDRMAP11.
Programming Mode: Static

Bits 11:8 – ADDRMAP_ROW_B1[3:0] Selects the HIF address bits used as row address bit 1.
Valid Range: 0 to 11
Internal Base: 7
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.
Programming Mode: Static

Bits 3:0 – ADDRMAP_ROW_B0[3:0] Selects the HIF address bits used as row address bit 0.
Valid Range: 0 to 11
Internal Base: 6
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.
Programming Mode: Static

17.6.53 UDDRC Address Map Register 6

Name: UDDRC_ADDRMAP6
Offset: 0x218
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	LPDDR3_6GB_12GB				ADDRMAP_ROW_B15[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
					ADDRMAP_ROW_B14[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					ADDRMAP_ROW_B13[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					ADDRMAP_ROW_B12[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – LPDDR3_6GB_12GB
Set this to 1 if there is an LPDDR3 SDRAM 6Gb or 12Gb device in use.
Present only in designs configured to support LPDDR3.
Programming Mode: Static

Value	Description
0	Non-LPDDR3 6Gb/12Gb device in use. All addresses are valid.
1	LPDDR3 SDRAM 6Gb/12Gb device in use. Every address having row[14:13]==2'b11 is considered as invalid.

Bits 27:24 – ADDRMAP_ROW_B15[3:0] Selects the HIF address bit used as row address bit 15.
Valid Range: 0 to 11, and 15
Internal Base: 21
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then row address bit 15 is set to 0.
Programming Mode: Static

Bits 19:16 – ADDRMAP_ROW_B14[3:0] Selects the HIF address bit used as row address bit 14.
Valid Range: 0 to 11, and 15
Internal Base: 20
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then row address bit 14 is set to 0.
Programming Mode: Static

Bits 11:8 – ADDRMAP_ROW_B13[3:0] Selects the HIF address bit used as row address bit 13.
Valid Range: 0 to 11, and 15
Internal Base: 19
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then row address bit 13 is set to 0.
Programming Mode: Static

Bits 3:0 – ADDRMAP_ROW_B12[3:0] Selects the HIF address bit used as row address bit 12.
Valid Range: 0 to 11, and 15
Internal Base: 18
The selected HIF address bit is determined by adding the internal base to the value of this field.
If unused, set to 15 and then row address bit 12 is set to 0.
Programming Mode: Static

17.6.54 UDDRC Address Map Register 9

Name: UDDRC_ADDRMAP9
Offset: 0x224
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					ADDRMAP_ROW_B5[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					ADDRMAP_ROW_B4[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					ADDRMAP_ROW_B3[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					ADDRMAP_ROW_B2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – ADDRMAP_ROW_B5[3:0] Selects the HIF address bits used as row address bit 5.
Valid Range: 0 to 11
Internal Base: 11
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 19:16 – ADDRMAP_ROW_B4[3:0] Selects the HIF address bits used as row address bit 4.
Valid Range: 0 to 11
Internal Base: 10
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 11:8 – ADDRMAP_ROW_B3[3:0] Selects the HIF address bits used as row address bit 3.
Valid Range: 0 to 11
Internal Base: 9
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 3:0 – ADDRMAP_ROW_B2[3:0] Selects the HIF address bits used as row address bit 2.
Valid Range: 0 to 11
Internal Base: 8
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

17.6.55 UDDRC Address Map Register 10

Name: UDDRC_ADDRMAP10
Offset: 0x228
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					ADDRMAP_ROW_B9[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					ADDRMAP_ROW_B8[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					ADDRMAP_ROW_B7[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					ADDRMAP_ROW_B6[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – ADDRMAP_ROW_B9[3:0] Selects the HIF address bits used as row address bit 9.
Valid Range: 0 to 11
Internal Base: 15
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 19:16 – ADDRMAP_ROW_B8[3:0] Selects the HIF address bits used as row address bit 8.
Valid Range: 0 to 11
Internal Base: 14
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 11:8 – ADDRMAP_ROW_B7[3:0] Selects the HIF address bits used as row address bit 7.
Valid Range: 0 to 11
Internal Base: 13
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

Bits 3:0 – ADDRMAP_ROW_B6[3:0] Selects the HIF address bits used as row address bit 6.
Valid Range: 0 to 11
Internal Base: 12
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

17.6.56 UDDRC Address Map Register 11

Name: UDDRC_ADDRMAP11
Offset: 0x22C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ADDRMAP_ROW_B10[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ADDRMAP_ROW_B10[3:0] Selects the HIF address bits used as row address bit 10.
Valid Range: 0 to 11
Internal Base: 16
The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
Programming Mode: Static

17.6.57 UDDRC ODT Configuration Register

Name: UDDRC_ODTCFG
Offset: 0x240
Reset: 0x04000400
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					WR_ODT_HOLD[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	23	22	21	20	19	18	17	16
				WR_ODT_DELAY[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					RD_ODT_HOLD[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	7	6	5	4	3	2	1	0
		RD_ODT_DELAY[4:0]						
Access		R/W	R/W	R/W	R/W	R/W		
Reset		0	0	0	0	0		

Bits 27:24 – WR_ODT_HOLD[3:0]
DFI PHY clock cycles to hold ODT for a write command. The minimum supported value is 2.
Recommended values:
DDR2:
- BL8: 0x5 (DDR2-400/533/667), 0x6 (DDR2-800), 0x7 (DDR2-1066)
- BL4: 0x3 (DDR2-400/533/667), 0x4 (DDR2-800), 0x5 (DDR2-1066)
DDR3:
- BL8: 0x6
WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble)
CRC_MODE = 0 (not CRC mode), 1 (CRC mode)
LPDDR3:
- BL8: 7 + RU(tODTon(max)/tCK)
Unit: DFI PHY clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 4

Bits 20:16 – WR_ODT_DELAY[4:0]
The delay, in DFI PHY clock cycles, from issuing a write command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the UDDRC.
Recommended values:
DDR2:
- CWL + AL - 3 (DDR2-400/533/667), CWL + AL - 4 (DDR2-800), CWL + AL - 5 (DDR2-1066)
If (CWL + AL - 3 < 0), UDDRC does not support ODT for write operation.
DDR3:
- 0x0
LPDDR3:
- WL - 1 - RU(tODTon(max)/tCK))
Unit: DFI PHY clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 4

Bits 11:8 – RD_ODT_HOLD[3:0]
DFI PHY clock cycles to hold ODT for a read command. The minimum supported value is 2.
Recommended values:
DDR2:
- BL8: 0x6 (not DDR2-1066), 0x7 (DDR2-1066)
- BL4: 0x4 (not DDR2-1066), 0x5 (DDR2-1066)
DDR3:
- BL8 - 0x6
LPDDR3:
- BL8: 5 + RU(tDQSCK(max)/tCK) - RD(tDQSCK(min)/tCK) + RU(tODTon(max)/tCK)
Unit: DFI PHY clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 4

Bits 6:2 – RD_ODT_DELAY[4:0]
The delay, in DFI PHY clock cycles, from issuing a read command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the UDDRC.
Recommended values:
DDR2:
- CL + AL - 4 (not DDR2-1066), CL + AL - 5 (DDR2-1066)
If (CL + AL - 4 < 0), UDDRC does not support ODT for read operation.
DDR3:
- CL - CWL
LPDDR3:
- RL + RD(tDQSCK(min)/tCK) - 1 - RU(tODTon(max)/tCK)
Unit: DFI PHY clock cycles.
Programming Mode: Quasi-dynamic Group 1, Group 4

17.6.58 UDDRC ODT/Rank Map Register

Name: UDDRC_ODTMAP
Offset: 0x244
Reset: 0x00000011
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				RANK0_RD_ODT				RANK0_WR_ODT
Access				R/W				R/W
Reset				1				1

Bit 4 – RANK0_RD_ODT Indicates which remote ODTs must be turned on during a read from rank 0.
Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here.
Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc.
For each rank, set its bit to 1 to enable its ODT.
Programming Mode: Static

Bit 0 – RANK0_WR_ODT Indicates which remote ODTs must be turned on during a write to rank 0.
Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here.
Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc.
For each rank, set its bit to 1 to enable its ODT.
Programming Mode: Static

17.6.59 UDDRC Scheduler Control Register 0

Name: UDDRC_SCHED
Offset: 0x250
Reset: 0x00001005
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		RDWR_IDLE_GAP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GO2CRITICAL_HYSTERESIS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				LPR_NUM_ENTRIES[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						PAGECLOSE	PREFER_WRITE	FORCE_LOW_PRI_N
Access						R/W	R/W	R/W
Reset						1	0	1

Bits 30:24 – RDWR_IDLE_GAP[6:0] When the preferred transaction store is empty for these many clock cycles, switch to the alternate transaction store if it is non-empty.
The read transaction store (both high and low priority) is the default preferred transaction store and the write transaction store is the alternative store.
When prefer write over read is set this is reversed.
0x0 is a legal value for this register. When set to 0x0, the transaction store switching will happen immediately when the switching conditions become true.
For performance only.
Unit: DFI clock cycles.
Programming Mode: Static

Bits 23:16 – GO2CRITICAL_HYSTERESIS[7:0] UNUSED
Programming Mode: Static

Bits 12:8 – LPR_NUM_ENTRIES[4:0] Number of entries in the low priority transaction store is this value + 1.
(2 - (SCHED.lpr_num_entries + 1)) is the number of entries available for the high priority transaction store.
Setting this to maximum value allocates all entries to low priority transaction store.
Setting this to 0 allocates 1 entry to low priority transaction store and the rest to high priority transaction store.
Programming Mode: Static

Bit 2 – PAGECLOSE If true, bank is kept open only while there are page hit transactions available in the CAM to that bank. The last read or write command in the CAM with a bank and page hit will be executed with auto-precharge if SCHED1.pageclose_timer=0. Even if this register set to 1 and SCHED1.pageclose_timer is set to 0, explicit precharge (and not auto-precharge) may be issued in some cases where there is a mode switch between Write and Read or between LPR and HPR.
If false, the bank remains open until there is a need to close it (to open a different page, or for page timeout or refresh timeout) - also known as open page policy. The open page policy can be overridden by setting the per-command-autopre bit on the HIF interface (hif_cmd_autopre).
The pageclose feature provids a midway between Open and Close page policies.

For performance only.
Programming Mode: Quasi-dynamic Group 3

Bit 1 – PREFER_WRITE If set then the bank selector prefers writes over reads.
For debug only.
Programming Mode: Static

Bit 0 – FORCE_LOW_PRI_N Active low signal. When asserted ('0'), all incoming transactions are forced to low priority. This implies that all High Priority Read (HPR) and Variable Priority Read commands (VPR) will be treated as Low Priority Read (LPR) commands. On the write side, all Variable Priority Write (VPW) commands will be treated as Normal Priority Write (NPW) commands. Forcing the incoming transactions to low priority implicitly turns off Bypass path for read commands.
For performance only.
Programming Mode: Static

17.6.60 UDDRC Scheduler Control Register 1

Name: UDDRC_SCHED1
Offset: 0x254
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PAGECLOSE_TIMER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PAGECLOSE_TIMER[7:0] This field works in conjunction with UDDRC_SCHED.PAGECLOSE.
It only has meaning if UDDRC_SCHED.PAGECLOSE=1.
If UDDRC_SCHED.PAGECLOSE=1 and UDDRC_SCHED1.PAGECLOSE=0, then an auto-precharge may be scheduled for last read or write command in the CAM with a bank and page hit.
Note, sometimes an explicit precharge is scheduled instead of the auto-precharge. See UDDRC_SCHED.PAGECLOSE for details of when this may happen.
If UDDRC_SCHED.PAGECLOSE=1 and PAGECLOSE_TIMER>0, then an auto-precharge is not scheduled for last read or write command in the CAM with a bank and page hit.
Instead, a timer is started, with PAGECLOSE_TIMER as the initial value.
There is a timer on a per bank basis.
The timer decrements unless the next read or write in the CAM to a bank is a page hit.
It gets reset to PAGECLOSE_TIMER value if the next read or write in the CAM to a bank is a page hit.
Once the timer has reached zero, an explicit precharge will be attempted to be scheduled.
Unit: DFI clock cycles.
Programming Mode: Static

17.6.61 UDDRC High Priority Read CAM Register 1

Name: UDDRC_PERFHPR1
Offset: 0x25C
Reset: 0x0F000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HPR_XACT_RUN_LENGTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	HPR_MAX_STARVE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HPR_MAX_STARVE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:24 – HPR_XACT_RUN_LENGTH[7:0]
Number of transactions that are serviced once the HPR queue goes critical is the smaller of:
- (a) This number
- (b) Number of transactions available.

Unit: Transaction.
For performance only.
Programming Mode: Quasi-dynamic Group 3

Bits 15:0 – HPR_MAX_STARVE[15:0] Number of DFI clocks that the HPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality; during normal operation, this function should not be disabled as it will cause excessive latencies.
For performance only.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 3

17.6.62 UDDRC Low Priority Read CAM Register 1

Name: UDDRC_PERFLPR1
Offset: 0x264
Reset: 0x0F00007F
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	LPR_XACT_RUN_LENGTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LPR_MAX_STARVE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPR_MAX_STARVE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	1	1	1	1

Bits 31:24 – LPR_XACT_RUN_LENGTH[7:0]
Number of transactions that are serviced once the LPR queue goes critical is the smaller of:
- (a) This number
- (b) Number of transactions available.

Unit: Transaction.
For performance only.
Programming Mode: Quasi-dynamic Group 3

Bits 15:0 – LPR_MAX_STARVE[15:0] Number of DFI clocks that the LPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality; during normal operation, this function should not be disabled as it will cause excessive latencies.
For performance only.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 3

17.6.63 UDDRC Write CAM Register 1

Name: UDDRC_PERFWR1
Offset: 0x26C
Reset: 0x0F00007F
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	W_XACT_RUN_LENGTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	W_MAX_STARVE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	W_MAX_STARVE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	1	1	1	1

Bits 31:24 – W_XACT_RUN_LENGTH[7:0]
Number of transactions that are serviced once the WR queue goes critical is the smaller of:
- (a) This number
- (b) Number of transactions available.

Unit: Transaction.
For performance only.
Programming Mode: Quasi-dynamic Group 3

Bits 15:0 – W_MAX_STARVE[15:0] Number of DFI clocks that the WR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality; during normal operation, this function should not be disabled as it will cause excessive latencies.
For performance only.
Unit: DFI clock cycles.
Programming Mode: Quasi-dynamic Group 3

17.6.64 UDDRC Debug Register 0

Name: UDDRC_DBG0
Offset: 0x300
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				DIS_COLLISION_PAGE_OPT		DIS_ACT_BYPASS	DIS_RD_BYPASS	DIS_WC
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit 4 – DIS_COLLISION_PAGE_OPT When this is set to '0', auto-precharge is disabled for the flushed command in a collision case. Collision cases are write followed by read to same address, read followed by write to same address, or write followed by write to same address with DBG0.dis_wc bit = 1 (where same address comparisons exclude the two address bits representing critical word).
For debug only.
Programming Mode: Static

Bit 2 – DIS_ACT_BYPASS Only present in designs supporting activate bypass.
When 1, disable bypass path for high priority read activates
For debug only.
Programming Mode: Static

Bit 1 – DIS_RD_BYPASS Only present in designs supporting read bypass.
When 1, disable bypass path for high priority read page hits.
For debug only.
Programming Mode: Static

Bit 0 – DIS_WC When 1, disable write combine.
For debug only
Programming Mode: Static

17.6.65 UDDRC Debug Register 1

Name: UDDRC_DBG1
Offset: 0x304
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DIS_HIF	DIS_DQ
Access							R/W	R/W
Reset							0	0

Bit 1 – DIS_HIF When 1, UDDRC asserts the HIF command signal hif_cmd_stall. UDDRC will ignore the hif_cmd_valid and all other associated request signals.
This bit is intended to be switched on-the-fly.
Programming Mode: Dynamic

Bit 0 – DIS_DQ When 1, UDDRC will not de-queue any transactions from the CAM. Bypass is also disabled. All transactions are queued in the CAM. No reads or writes are issued to SDRAM as long as this is asserted.
This bit may be used to prevent reads or writes being issued by the UDDRC, which makes it safe to modify certain register fields associated with reads and writes (see User Guide for details). After setting this bit, it is strongly recommended to poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty, before making changes to any registers which affect reads and writes. This will ensure that the relevant logic in the DDRC is idle.
This bit is intended to be switched on-the-fly.
Programming Mode: Dynamic

17.6.66 UDDRC CAM Debug Register

Name: UDDRC_DBGCAM
Offset: 0x308
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			WR_DATA_PIPELINE_EMPTY	RD_DATA_PIPELINE_EMPTY		DBG_WR_Q_EMPTY	DBG_RD_Q_EMPTY	DBG_STALL
Access			R	R		R	R	R
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
			DBG_W_Q_DEPTH[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			DBG_LPR_Q_DEPTH[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			DBG_HPR_Q_DEPTH[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 29 – WR_DATA_PIPELINE_EMPTY This bit indicates that the write data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/ data have completed.
Programming Mode: Dynamic

Bit 28 – RD_DATA_PIPELINE_EMPTY This bit indicates that the read data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/ data have completed.
Programming Mode: Dynamic

Bit 26 – DBG_WR_Q_EMPTY When 1, all the Write command queues and Write data buffers inside DDRC are empty. This register is to be used for debug purpose.
An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time.
For debug only
Programming Mode: Dynamic

Bit 25 – DBG_RD_Q_EMPTY When 1, all the Read command queues and Read data buffers inside DDRC are empty. This register is to be used for debug purpose.
An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time.
For debug only
Programming Mode: Dynamic

Bit 24 – DBG_STALL Stall
For debug only
Programming Mode: Dynamic

Bits 21:16 – DBG_W_Q_DEPTH[5:0] Write queue depth
This entry is not included in the calculation of the queue depth.
For debug only
Programming Mode: Dynamic

Bits 13:8 – DBG_LPR_Q_DEPTH[5:0] Low priority read queue depth
This entry is not included in the calculation of the queue depth.
For debug only
Programming Mode: Dynamic

Bits 5:0 – DBG_HPR_Q_DEPTH[5:0] High priority read queue depth
For debug only
Programming Mode: Dynamic

17.6.67 UDDRC Command Debug Register

Name: UDDRC_DBGCMD
Offset: 0x30C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			CTRLUPD	ZQ_CALIB_SHORT				RANK0_REFRESH
Access			R/W	R/W				R/W
Reset			0	0				0

Bit 5 – CTRLUPD
Setting this register bit to 1 indicates to the UDDRC to issue a dfi_ctrlupd_req to the PHY. When this request is stored in the UDDRC, the bit is automatically cleared. This operation must only be performed when DFIUPD0.dis_auto_ctrlupd=1.
Programming Mode: Dynamic

Bit 4 – ZQ_CALIB_SHORT
Setting this register bit to 1 indicates to the UDDRC to issue a ZQCS (ZQ calibration short)/MPC(ZQ calibration) command to the SDRAM. When this request is stored in the UDDRC, the bit is automatically cleared. This operation can be performed only when ZQCTL0.dis_auto_zq=1. It is recommended NOT to set this register bit if in Init, in Self-Refresh or Deep power-down operating modes or Maximum Power Saving Mode.
For Self-Refresh it will be scheduled after SR has been exited.
For Deep power down and Maximum Power Saving Mode, it will not be scheduled, although DBGSTAT.zq_calib_short_busy will be de-asserted.
Programming Mode: Dynamic

Bit 0 – RANK0_REFRESH
Setting this register bit to 1 indicates to the UDDRC to issue a refresh to rank 0. Writing to this bit causes DBGSTAT.rank0_refresh_busy to be set. When DBGSTAT.rank0_refresh_busy is cleared, the command has been stored in UDDRC.
For 3DS configuration, refresh is sent to rank index 0.
This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode.
Programming Mode: Dynamic

17.6.68 UDDRC Status Debug Register

Name: UDDRC_DBGSTAT
Offset: 0x310
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CTRLUPD_BUSY	ZQ_CALIB_SHORT_BUSY				RANK0_REFRESH_BUSY
Access			R	R				R
Reset			0	0				0

Bit 5 – CTRLUPD_BUSY SoC core may initiate a ctrlupd operation only if this signal is low.
This signal goes high in the clock after the UDDRC accepts the ctrlupd request. It goes low when the ctrlupd operation is initiated in the UDDRC.
It is recommended not to perform ctrlupd operations when this signal is high.
Programming Mode: Dynamic

Value	Description
0	Indicates that the SoC core can initiate a ctrlupd operation.
1	Indicates that ctrlupd operation has not been initiated yet in the UDDRC.

Bit 4 – ZQ_CALIB_SHORT_BUSY SoC core may initiate a ZQCS (ZQ calibration short) operation only if this signal is low. This signal goes high in the clock after the UDDRC accepts the ZQCS request. It goes low when the ZQCS operation is initiated in the UDDRC. It is recommended not to perform ZQCS operations when this signal is high.
Programming Mode: Dynamic

Value	Description
0	Indicates that the SoC core can initiate a ZQCS operation.
1	Indicates that ZQCS operation has not been initiated yet in the UDDRC.

Bit 0 – RANK0_REFRESH_BUSY SoC core may initiate a rank0_refresh operation (refresh operation to rank 0) only if this signal is low.
This signal goes high in the clock after DBGCMD.rank0_refresh is set to one. It goes low when the rank0_refresh operation is stored in the UDDRC.
It is recommended not to perform rank0_refresh operations when this signal is high.
Programming Mode: Dynamic

Value	Description
0	Indicates that the SoC core can initiate a rank0_refresh operation.
1	Indicates that rank0_refresh operation has not been stored yet in the UDDRC.

17.6.69 UDDRC Software Register Programming Control Enable

Name: UDDRC_SWCTL
Offset: 0x320
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SW_DONE
Access								R/W
Reset								1

Bit 0 – SW_DONE Enable quasi-dynamic register programming outside reset. Program register to 0 to enable quasi-dynamic programming. Set back register to 1 once programming is done.
Programming Mode: Dynamic

17.6.70 UDDRC Software Register Programming Control Status

Name: UDDRC_SWSTAT
Offset: 0x324
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								SW_DONE_ACK
Access								R
Reset								1

Bit 0 – SW_DONE_ACK Register programming done. This register is the echo of SWCTL.sw_done. Wait for sw_done value 1 to propagate to sw_done_ack at the end of the programming sequence to ensure that the correct registers values are propagated to the destination clock domains.
Programming Mode: Static

17.6.71 UDDRC AXI Poison Configuration Register. Common for all AXI ports

Name: UDDRC_POISONCFG
Offset: 0x36C
Reset: 0x00110011
Property: R/W

Bit	31	30	29	28	27	26	25	24
								RD_POISON_INTR_CLR
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
				RD_POISON_INTR_EN				RD_POISON_SLVERR_EN
Access				R/W				R/W
Reset				1				1

Bit	15	14	13	12	11	10	9	8
								WR_POISON_INTR_CLR
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
				WR_POISON_INTR_EN				WR_POISON_SLVERR_EN
Access				R/W				R/W
Reset				1				1

Bit 24 – RD_POISON_INTR_CLR Interrupt clear for read transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts. UDDRC automatically clears this bit.
Programming Mode: Dynamic

Bit 20 – RD_POISON_INTR_EN If set to 1, enables interrupts for read transaction poisoning
Programming Mode: Dynamic

Bit 16 – RD_POISON_SLVERR_EN If set to 1, enables SLVERR response for read transaction poisoning
Programming Mode: Dynamic

Bit 8 – WR_POISON_INTR_CLR Interrupt clear for write transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts. UDDRC automatically clears this bit.
Programming Mode: Dynamic

Bit 4 – WR_POISON_INTR_EN If set to 1, enables interrupts for write transaction poisoning
Programming Mode: Dynamic

Bit 0 – WR_POISON_SLVERR_EN If set to 1, enables SLVERR response for write transaction poisoning
Programming Mode: Dynamic

17.6.72 UDDRC AXI Poison Status Register

Name: UDDRC_POISONSTAT
Offset: 0x370
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				RD_POISON_INTR_4	RD_POISON_INTR_3	RD_POISON_INTR_2	RD_POISON_INTR_1	RD_POISON_INTR_0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				WR_POISON_INTR_4	WR_POISON_INTR_3	WR_POISON_INTR_2	WR_POISON_INTR_1	WR_POISON_INTR_0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 20 – RD_POISON_INTR_4 Read transaction poisoning error interrupt for port 4. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 19 – RD_POISON_INTR_3 Read transaction poisoning error interrupt for port 3. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 18 – RD_POISON_INTR_2 Read transaction poisoning error interrupt for port 2. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 17 – RD_POISON_INTR_1 Read transaction poisoning error interrupt for port 1. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 16 – RD_POISON_INTR_0 Read transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding

AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 4 – WR_POISON_INTR_4 Write transaction poisoning error interrupt for port 4. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 3 – WR_POISON_INTR_3 Write transaction poisoning error interrupt for port 3. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 2 – WR_POISON_INTR_2 Write transaction poisoning error interrupt for port 2. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 1 – WR_POISON_INTR_1 Write transaction poisoning error interrupt for port 1. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

Bit 0 – WR_POISON_INTR_0 Write transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.
Programming Mode: Dynamic

17.6.73 UDDRC Temperature Derate Status Register

Name: UDDRC_DERATESTAT
Offset: 0x3F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DERATE_TEMP_LIMIT_INTR
Access								R
Reset								0

Bit 0 – DERATE_TEMP_LIMIT_INTR Derate temperature interrupt indicating LPDDR2/3 SDRAM temperature operating limit is exceeded.
This register field is set to 1 when the value read from MR4[2:0] is 3'b000 or 3'b111. Cleared by register DERATECTL.derate_temp_limit_intr_clr.
Programming Mode: Static

17.6.74 UDDRC Port Status Register

Name: UDDRC_PSTAT
Offset: 0x3FC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				WR_PORT_BUSY_4	WR_PORT_BUSY_3	WR_PORT_BUSY_2	WR_PORT_BUSY_1	WR_PORT_BUSY_0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				RD_PORT_BUSY_4	RD_PORT_BUSY_3	RD_PORT_BUSY_2	RD_PORT_BUSY_1	RD_PORT_BUSY_0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 20 – WR_PORT_BUSY_4 Indicates if there are outstanding writes for AXI port 4.
Programming Mode: Dynamic

Bit 19 – WR_PORT_BUSY_3 Indicates if there are outstanding writes for AXI port 3.
Programming Mode: Dynamic

Bit 18 – WR_PORT_BUSY_2 Indicates if there are outstanding writes for AXI port 2.
Programming Mode: Dynamic

Bit 17 – WR_PORT_BUSY_1 Indicates if there are outstanding writes for AXI port 1.
Programming Mode: Dynamic

Bit 16 – WR_PORT_BUSY_0 Indicates if there are outstanding writes for AXI port 0.
Programming Mode: Dynamic

Bit 4 – RD_PORT_BUSY_4 Indicates if there are outstanding reads for AXI port 4.
Programming Mode: Dynamic

Bit 3 – RD_PORT_BUSY_3 Indicates if there are outstanding reads for AXI port 3.
Programming Mode: Dynamic

Bit 2 – RD_PORT_BUSY_2 Indicates if there are outstanding reads for AXI port 2.
Programming Mode: Dynamic

Bit 1 – RD_PORT_BUSY_1 Indicates if there are outstanding reads for AXI port 1.
Programming Mode: Dynamic

Bit 0 – RD_PORT_BUSY_0 Indicates if there are outstanding reads for AXI port 0.
Programming Mode: Dynamic

17.6.75 UDDRC Port Common Configuration Register

Name: UDDRC_PCCFG
Offset: 0x400
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								BL_EXP_MODE
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
				PAGEMATCH_LIMIT				GO2CRITICAL_EN
Access				R/W				R/W
Reset				0				0

Bit 8 – BL_EXP_MODE Burst length expansion mode. By default (i.e. bl_exp_mode==0) XPI expands every AXI burst into multiple HIF commands, using the memory burst length as a unit. If set to 1, then XPI will use half of the memory burst length as a unit.
This applies to both reads and writes. When MSTR.data_bus_width==00, setting bl_exp_mode to 1 has no effect.
Functionality is also not supported if Data Channel Interleave is enabled
Programming Mode: Static

Bit 4 – PAGEMATCH_LIMIT Page match four limit. If set to 1, limits the number of consecutive same page DDRC transactions that can be granted by the Port Arbiter to four when Page Match feature is enabled. If set to 0, there is no limit imposed on number of consecutive same page DDRC transactions.
Programming Mode: Static

Bit 0 – GO2CRITICAL_EN If set to 1 (enabled), sets the co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals going to DDRC based on urgent input (awurgent, arurgent) coming from the AXI host. If set to 0 (disabled), the co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals at DDRC are driven to 1b'0.
For uPCTL2, this register field must be set to 0
Programming Mode: Static

17.6.76 UDDRC AXI Port 0 Configuration Read Register

Name: UDDRC_PCFGR_0
Offset: 0x404
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RD_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – RD_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – RD_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_lpr/ co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – RD_PORT_AGING_EN If set to 1, enables aging function for the read channel of the port.
Programming Mode: Static

Bits 9:0 – RD_PORT_PRIORITY[9:0]
Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.77 UDDRC AXI Port 0 Configuration Write Register

Name: UDDRC_PCFGW_0
Offset: 0x408
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	WR_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WR_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – WR_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register.
Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – WR_PORT_AGING_EN If set to 1, enables aging function for the write channel of the port.
Programming Mode: Static

Bits 9:0 – WR_PORT_PRIORITY[9:0]
Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level.
For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable).
For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.78 UDDRC AXI Port 0 Control Register

Name: UDDRC_PCTRL_0
Offset: 0x490
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_EN
Access								R/W
Reset								0

Bit 0 – PORT_EN Enables AXI port 0.
Programming Mode: Dynamic

17.6.79 UDDRC AXI Port 0 Read QoS Configuration Register 0

Name: UDDRC_PCFGQOS0_0
Offset: 0x494
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 21:20 – RQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
- 0 : LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region1 maps to the blue address queue.
In this case, valid values are
- 0: LPR
- 1: VPR only
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – RQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
- 0: LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region 0 maps to the blue address queue.
In this case, valid values are:
0: LPR and 1: VPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – RQOS_MAP_LEVEL1[3:0] Separation level1 indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.

Programming Mode: Quasi-dynamic Group 3

17.6.80 UDDRC AXI Port 0 Read QoS Configuration Register 1

Name: UDDRC_PCFGQOS1_0
Offset: 0x498
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RQOS_MAP_TIMEOUTR[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	RQOS_MAP_TIMEOUTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						RQOS_MAP_TIMEOUTB[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	RQOS_MAP_TIMEOUTB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RQOS_MAP_TIMEOUTR[10:0] Specifies the timeout value for transactions mapped to the red address queue.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – RQOS_MAP_TIMEOUTB[10:0] Specifies the timeout value for transactions mapped to the blue address queue.
Programming Mode: Quasi-dynamic Group 3

17.6.81 UDDRC AXI Port 0 Write QoS Configuration Register 0

Name: UDDRC_PCFGWQOS0_0
Offset: 0x49C
Reset: 0x00000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							WQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					WQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					WQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – WQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region 2.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 2 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – WQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – WQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – WQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to awqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – WQOS_MAP_LEVEL1[3:0] Separation level indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 which corresponds to awqos.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.82 UDDRC AXI Port 0 Write QoS Configuration Register 1

Name: UDDRC_PCFGWQOS1_0
Offset: 0x4A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						WQOS_MAP_TIMEOUT2[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	WQOS_MAP_TIMEOUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						WQOS_MAP_TIMEOUT1[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WQOS_MAP_TIMEOUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – WQOS_MAP_TIMEOUT2[10:0] Specifies the timeout value for write transactions in region 2.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – WQOS_MAP_TIMEOUT1[10:0] Specifies the timeout value for write transactions in region 0 and 1.
Programming Mode: Quasi-dynamic Group 3

17.6.83 UDDRC AXI Port 1 Configuration Read Register

Name: UDDRC_PCFGR_1
Offset: 0x4B4
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RD_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – RD_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – RD_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – RD_PORT_AGING_EN If set to 1, enables aging function for the read channel of the port.
Programming Mode: Static

Bits 9:0 – RD_PORT_PRIORITY[9:0]
Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.84 UDDRC AXI Port 1 Configuration Write Register

Name: UDDRC_PCFGW_1
Offset: 0x4B8
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	WR_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WR_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – WR_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register.
Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – WR_PORT_AGING_EN If set to 1, enables aging function for the write channel of the port.
Programming Mode: Static

Bits 9:0 – WR_PORT_PRIORITY[9:0] Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the
write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level.
For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable).
For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.85 UDDRC AXI Port 1 Control Register

Name: UDDRC_PCTRL_1
Offset: 0x540
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_EN
Access								R/W
Reset								0

Bit 0 – PORT_EN Enables AXI port 1.
Programming Mode: Dynamic

17.6.86 UDDRC AXI Port 1 Read QoS Configuration Register 0

Name: UDDRC_PCFGQOS0_1
Offset: 0x544
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					RQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 21:20 – RQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
- 0 : LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region1 maps to the blue address queue.
In this case, valid values are
- 0: LPR
- 1: VPR only
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – RQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
- 0: LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region 0 maps to the blue address queue.
In this case, valid values are:
0: LPR and 1: VPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – RQOS_MAP_LEVEL1[3:0] Separation level1 indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.

Programming Mode: Quasi-dynamic Group 3

17.6.87 UDDRC AXI Port 1 Read QoS Configuration Register 1

Name: UDDRC_PCFGQOS1_1
Offset: 0x548
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RQOS_MAP_TIMEOUTR[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	RQOS_MAP_TIMEOUTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						RQOS_MAP_TIMEOUTB[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	RQOS_MAP_TIMEOUTB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RQOS_MAP_TIMEOUTR[10:0] Specifies the timeout value for transactions mapped to the red address queue.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – RQOS_MAP_TIMEOUTB[10:0] Specifies the timeout value for transactions mapped to the blue address queue.
Programming Mode: Quasi-dynamic Group 3

17.6.88 UDDRC AXI Port 1 Write QoS Configuration Register 0

Name: UDDRC_PCFGWQOS0_1
Offset: 0x54C
Reset: 0x00000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							WQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					WQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					WQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – WQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region 2.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 2 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – WQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – WQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – WQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to awqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – WQOS_MAP_LEVEL1[3:0] Separation level indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 which corresponds to awqos.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.89 UDDRC AXI Port 1 Write QoS Configuration Register 1

Name: UDDRC_PCFGWQOS1_1
Offset: 0x550
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						WQOS_MAP_TIMEOUT2[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	WQOS_MAP_TIMEOUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						WQOS_MAP_TIMEOUT1[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WQOS_MAP_TIMEOUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – WQOS_MAP_TIMEOUT2[10:0] Specifies the timeout value for write transactions in region 2.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – WQOS_MAP_TIMEOUT1[10:0] Specifies the timeout value for write transactions in region 0 and 1.
Programming Mode: Quasi-dynamic Group 3

17.6.90 UDDRC AXI Port 2 Configuration Read Register

Name: UDDRC_PCFGR_2
Offset: 0x564
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RD_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – RD_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – RD_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – RD_PORT_AGING_EN If set to 1, enables aging function for the read channel of the port.
Programming Mode: Static

Bits 9:0 – RD_PORT_PRIORITY[9:0]
Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.91 UDDRC AXI Port 2 Configuration Write Register

Name: UDDRC_PCFGW_2
Offset: 0x568
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	WR_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WR_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – WR_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register.
Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – WR_PORT_AGING_EN If set to 1, enables aging function for the write channel of the port.
Programming Mode: Static

Bits 9:0 – WR_PORT_PRIORITY[9:0]
Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level.
For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable).
For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.92 UDDRC AXI Port 2 Control Register

Name: UDDRC_PCTRL_2
Offset: 0x5F0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_EN
Access								R/W
Reset								0

Bit 0 – PORT_EN Enables AXI port 2.
Programming Mode: Dynamic

17.6.93 UDDRC AXI Port 2 Read QoS Configuration Register 0

Name: UDDRC_PCFGQOS0_2
Offset: 0x5F4
Reset: 0x02000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							RQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							1	0

Bit	23	22	21	20	19	18	17	16
			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					RQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					RQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – RQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region2.
For dual address queue configurations, region2 maps to the red address queue.
Valid values are 1: VPR and 2: HPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region2 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – RQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
- 0 : LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region1 maps to the blue address queue.
In this case, valid values are
- 0: LPR
- 1: VPR only
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – RQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
- 0: LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region 0 maps to the blue address queue.
In this case, valid values are:
0: LPR and 1: VPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.

Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – RQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to arqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – RQOS_MAP_LEVEL1[3:0] Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.94 UDDRC AXI Port 2 Read QoS Configuration Register 1

Name: UDDRC_PCFGQOS1_2
Offset: 0x5F8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RQOS_MAP_TIMEOUTR[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	23	22	21	20	19	18	17	16
	RQOS_MAP_TIMEOUTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						RQOS_MAP_TIMEOUTB[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
	RQOS_MAP_TIMEOUTB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RQOS_MAP_TIMEOUTR[10:0] Specifies the timeout value for transactions mapped to the red address queue.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – RQOS_MAP_TIMEOUTB[10:0] Specifies the timeout value for transactions mapped to the blue address queue.
Programming Mode: Quasi-dynamic Group 3

17.6.95 UDDRC AXI Port 2 Write QoS Configuration Register 0

Name: UDDRC_PCFGWQOS0_2
Offset: 0x5FC
Reset: 0x00000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							WQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					WQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					WQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – WQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region 2.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 2 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – WQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – WQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – WQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to awqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – WQOS_MAP_LEVEL1[3:0] Separation level indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 which corresponds to awqos.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.96 UDDRC AXI Port 2 Write QoS Configuration Register 1

Name: UDDRC_PCFGWQOS1_2
Offset: 0x600
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						WQOS_MAP_TIMEOUT2[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	WQOS_MAP_TIMEOUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						WQOS_MAP_TIMEOUT1[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WQOS_MAP_TIMEOUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – WQOS_MAP_TIMEOUT2[10:0] Specifies the timeout value for write transactions in region 2.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – WQOS_MAP_TIMEOUT1[10:0] Specifies the timeout value for write transactions in region 0 and 1.
Programming Mode: Quasi-dynamic Group 3

17.6.97 UDDRC AXI Port 3 Configuration Read Register

Name: UDDRC_PCFGR_3
Offset: 0x614
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RD_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – RD_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – RD_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_lpr/ co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – RD_PORT_AGING_EN If set to 1, enables aging function for the read channel of the port.
Programming Mode: Static

Bits 9:0 – RD_PORT_PRIORITY[9:0]
Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.98 UDDRC AXI Port 3 Configuration Write Register

Name: UDDRC_PCFGW_3
Offset: 0x618
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	WR_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WR_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – WR_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register.
Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – WR_PORT_AGING_EN If set to 1, enables aging function for the write channel of the port.
Programming Mode: Static

Bits 9:0 – WR_PORT_PRIORITY[9:0]
Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level.
For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable).
For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.99 UDDRC AXI Port 3 Control Register

Name: UDDRC_PCTRL_3
Offset: 0x6A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_EN
Access								R/W
Reset								0

Bit 0 – PORT_EN Enables AXI port 3.
Programming Mode: Dynamic

17.6.100 UDDRC AXI Port 3 Read QoS Configuration Register 0

Name: UDDRC_PCFGQOS0_3
Offset: 0x6A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 21:20 – RQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
- 0 : LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region1 maps to the blue address queue.
In this case, valid values are
- 0: LPR
- 1: VPR only
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – RQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
- 0: LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region 0 maps to the blue address queue.
In this case, valid values are:
0: LPR and 1: VPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – RQOS_MAP_LEVEL1[3:0] Separation level1 indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.

Programming Mode: Quasi-dynamic Group 3

17.6.101 UDDRC AXI Port 3 Read QoS Configuration Register 1

Name: UDDRC_PCFGQOS1_3
Offset: 0x6A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RQOS_MAP_TIMEOUTR[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	23	22	21	20	19	18	17	16
	RQOS_MAP_TIMEOUTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						RQOS_MAP_TIMEOUTB[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
	RQOS_MAP_TIMEOUTB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RQOS_MAP_TIMEOUTR[10:0] Specifies the timeout value for transactions mapped to the red address queue.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – RQOS_MAP_TIMEOUTB[10:0] Specifies the timeout value for transactions mapped to the blue address queue.
Programming Mode: Quasi-dynamic Group 3

17.6.102 UDDRC AXI Port 3 Write QoS Configuration Register 0

Name: UDDRC_PCFGWQOS0_3
Offset: 0x6AC
Reset: 0x00000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							WQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					WQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					WQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – WQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region 2.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 2 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – WQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – WQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – WQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to awqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – WQOS_MAP_LEVEL1[3:0] Separation level indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 which corresponds to awqos.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.103 UDDRC AXI Port 3 Write QoS Configuration Register 1

Name: UDDRC_PCFGWQOS1_3
Offset: 0x6B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						WQOS_MAP_TIMEOUT2[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	WQOS_MAP_TIMEOUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						WQOS_MAP_TIMEOUT1[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WQOS_MAP_TIMEOUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – WQOS_MAP_TIMEOUT2[10:0] Specifies the timeout value for write transactions in region 2.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – WQOS_MAP_TIMEOUT1[10:0] Specifies the timeout value for write transactions in region 0 and 1.
Programming Mode: Quasi-dynamic Group 3

17.6.104 UDDRC AXI Port 4 Configuration Read Register

Name: UDDRC_PCFGR_4
Offset: 0x6C4
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RD_PORT_PAGEMATCH_EN	RD_PORT_URGENT_EN	RD_PORT_AGING_EN			RD_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RD_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – RD_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – RD_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – RD_PORT_AGING_EN If set to 1, enables aging function for the read channel of the port.
Programming Mode: Static

Bits 9:0 – RD_PORT_PRIORITY[9:0]
Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.105 UDDRC AXI Port 4 Configuration Write Register

Name: UDDRC_PCFGW_4
Offset: 0x6C8
Reset: 0x00004000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WR_PORT_PAGEMATCH_EN	WR_PORT_URGENT_EN	WR_PORT_AGING_EN			WR_PORT_PRIORITY[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	WR_PORT_PRIORITY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WR_PORT_PAGEMATCH_EN If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also UDDRC_PCCFG.PAGEMATCH_LIMIT.
Programming Mode: Static

Bit 13 – WR_PORT_URGENT_EN If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the host, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register.
Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
Programming Mode: Static

Bit 12 – WR_PORT_AGING_EN If set to 1, enables aging function for the write channel of the port.
Programming Mode: Static

Bits 9:0 – WR_PORT_PRIORITY[9:0]
Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level.
For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable).
For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching.
Programming Mode: Static
Note: The two LSBs of this register field are tied internally to 2'b00.

17.6.106 UDDRC AXI Port 4 Control Register

Name: UDDRC_PCTRL_4
Offset: 0x750
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_EN
Access								R/W
Reset								0

Bit 0 – PORT_EN Enables AXI port 4.
Programming Mode: Dynamic

17.6.107 UDDRC AXI Port 4 Read QoS Configuration Register 0

Name: UDDRC_PCFGQOS0_4
Offset: 0x754
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RQOS_MAP_REGION1[1:0]				RQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 21:20 – RQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
- 0 : LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region1 maps to the blue address queue.
In this case, valid values are
- 0: LPR
- 1: VPR only
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – RQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
- 0: LPR
- 1: VPR
- 2: HPR
For dual address queue configurations, region 0 maps to the blue address queue.
In this case, valid values are:
0: LPR and 1: VPR only.
When VPR support is disabled (UDDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – RQOS_MAP_LEVEL1[3:0] Separation level1 indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos.
Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.

Programming Mode: Quasi-dynamic Group 3

17.6.108 UDDRC AXI Port 4 Read QoS Configuration Register 1

Name: UDDRC_PCFGQOS1_4
Offset: 0x758
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RQOS_MAP_TIMEOUTR[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	RQOS_MAP_TIMEOUTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						RQOS_MAP_TIMEOUTB[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	RQOS_MAP_TIMEOUTB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RQOS_MAP_TIMEOUTR[10:0] Specifies the timeout value for transactions mapped to the red address queue.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – RQOS_MAP_TIMEOUTB[10:0] Specifies the timeout value for transactions mapped to the blue address queue.
Programming Mode: Quasi-dynamic Group 3

17.6.109 UDDRC AXI Port 4 Write QoS Configuration Register 0

Name: UDDRC_PCFGWQOS0_4
Offset: 0x75C
Reset: 0x00000E00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							WQOS_MAP_REGION2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			WQOS_MAP_REGION1[1:0]				WQOS_MAP_REGION0[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					WQOS_MAP_LEVEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	0

Bit	7	6	5	4	3	2	1	0
					WQOS_MAP_LEVEL1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 25:24 – WQOS_MAP_REGION2[1:0] This bitfield indicates the traffic class of region 2.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 2 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 21:20 – WQOS_MAP_REGION1[1:0] This bitfield indicates the traffic class of region 1.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 17:16 – WQOS_MAP_REGION0[1:0] This bitfield indicates the traffic class of region 0.
Valid values are:
0: NPW, 1: VPW.
When VPW support is disabled (UDDRC_VPW_EN = 0) and traffic class of region 0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
Programming Mode: Quasi-dynamic Group 3

Bits 11:8 – WQOS_MAP_LEVEL2[3:0] Separation level2 indicating the end of region1 mapping; start of region1 is (level1 + 1). Possible values for level2 are (level1 + 1) to 14 which corresponds to awqos.
Region2 starts from (level2 + 1) up to 15.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

Bits 3:0 – WQOS_MAP_LEVEL1[3:0] Separation level indicating the end of region0 mapping; start of region0 is 0.
Possible values for level1 are 0 to 13 which corresponds to awqos.
Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.
All of the map_level* registers must be set to distinct values.
Programming Mode: Quasi-dynamic Group 3

17.6.110 UDDRC AXI Port 4 Write QoS Configuration Register 1

Name: UDDRC_PCFGWQOS1_4
Offset: 0x760
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						WQOS_MAP_TIMEOUT2[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	WQOS_MAP_TIMEOUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						WQOS_MAP_TIMEOUT1[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WQOS_MAP_TIMEOUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – WQOS_MAP_TIMEOUT2[10:0] Specifies the timeout value for write transactions in region 2.
Programming Mode: Quasi-dynamic Group 3

Bits 10:0 – WQOS_MAP_TIMEOUT1[10:0] Specifies the timeout value for write transactions in region 0 and 1.
Programming Mode: Quasi-dynamic Group 3

17.6.111 UDDRC SAR Base Address Register 0

Name: UDDRC_SARBASE0
Offset: 0xF04
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BASE_ADDR[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – BASE_ADDR[2:0] Base address for address region 0 specified as awaddr[31:29] and araddr[31:29]
Programming Mode: Static

17.6.112 UDDRC SAR Size Register 0

Name: UDDRC_SARSIZE0
Offset: 0xF08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	NBLOCKS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – NBLOCKS[7:0] Number of blocks for address region 0.
This register determines the total size of the region in multiples of 512 Mbytes. The register value is encoded as number of blocks = nblocks + 1. For example, if register is programmed to 0, region will have 1 block.
Programming Mode: Static

18. DDR/LPDDR Physical Interface (DDR3PHY)

18.1 Description

The DDR/LPDDR DDR3PHY operates the physical interface to JEDEC standard DDR2, DDR3, LPDDR2, LPDDR3 SDRAM memories up to 1066 Mbps data rates.

18.2 Embedded Characteristics

- Compatible with JEDEC Standard DDR2/DDR3/DDR3L/LPDDR2/LPDDR3 SDRAMs
- Operating Range from 100 MHz (200 Mbps) to 533 MHz (1066 Mbps) in DDR2/DDR3/LPDDR2/LPDDR3 Modes
- Operating Range from 0 to 200 MHz in Mobile DDR Mode
- x16 External Data Bus Width
- Programmable Output and ODT (On Die Termination) Impedance with Dynamic PVT Compensation
- Embedded Dynamic Drift Detection in the PHY to facilitate Dynamic Drift Compensation with the Controller
- Host and Client DLLs used for Precise Timing Management
- Lane-based Architecture (Byte Lane, Command Lane)

18.2.1 Supported PHY Data Rates

Table 18-1. Supported Modes and Data Rates

Memory Type	PHY Maximum Speed	Unit
DDR2	1066	Mbps
DDR3	1066	
DDR3L	1066	
LPDDR2	1066	
LPDDR3	1066	

18.2.2 Applicable Standards

The following JEDEC standards are applicable:

- JESD8-15 JEDEC Stub Series Terminated Logic for 1.8V (SSTL_18) Specification
- JESD79-2 JEDEC DDR2 SDRAM Specification
- JESD79-3 JEDEC DDR3 SDRAM Specification
- JESD209-2 JEDEC LPDDR2 SDRAM Specification
- JESD209-3 JEDEC LPDDR3 SDRAM Specification

18.3 Functional Description

18.3.1 Byte Lane PHY

The external memory components are designed to support byte lanes for optimal system timing. The partitioning of the data word into discrete byte lanes allows pin-to-pin skew to be managed across a much smaller group of signals than would typically be required. This eases signal track topology matching in the package and PCB environments.

The SDRAM contains data strobes associated with each 8 bits of data and there is a timing skew allowance between the main clock signal to the SDRAM and its data strobe inputs during a Write command (tDQSS):

- 8-bit memory components provide a single DQS.

- 16-bit memory components provide two DQSs, one for each 8 bits.

A byte lane consists of the following I/O slots:

- 8 data bits (D0-D7)
- Data strobe bits (DQS/DQSN)
- 1 data mask bit (DQM)

The ITMs (Interface Timing Modules) provide a mechanism for monitoring read timing drift, which can be used to adjust timing to maintain optimum system margins. Drift analysis and compensation are performed by the controller on a per-byte lane basis. Read DQS recognition and drift compensation are described further in the section “Universal DDR Memory Controller (UDDRC)”. The ITM components contain the functions to monitor DQS drift and permit timing adjustment; the controller provides the analysis and control for these functions. These functions operate dynamically for each data bit of every user-issued Read command. There are no overhead penalties in terms of channel bandwidth or utilization incurred by the use of these functions.

The memory interface (PHY) architecture is based on the concept of independent, but related, signal groups to provide the highest level of system timing performance. In order to maintain robust system timing, all clock and data signals relevant to a byte lane remain within that byte lane. These signals are not shared between other byte lanes or between a byte lane and a command lane.

A DLL macrocell consisting of a host DLL and two client DLLs (mirror delay lines) is used at each byte lane to facilitate optimal PHY timing for drive and capture of DDR data streams, and allows the lanes to be independent. The host DLL section provides outputs for DDR data stream creation to the SDRAMs and acts as a reference for the client delay line sections. The client delay line sections translate the incoming DQS/DQS_b into the center of the read data eye to maximize read system timing margins.

The user can fine-tune the DQS and DQ signal relationships to maximize the read system timing margin. The DLL includes adjustability of the client delay lines for the DQS and DQS_b signals, which provide byte-wide timing adjustments. The ITMs include adjustability of the read DQS/DQS_b strobe timing, which provides byte-wide timing adjustments. The ITMs include adjustability of the read DQ signal timing, which provides per-bit timing adjustability. To enable lane-independent timing adjustments, DLL adjustment bits are provided by the controller per byte lane, and ITM adjustment bits are provided per bit.

The DDR-specific SSTL I/Os include programmable ODT and output impedance selection. The ODT and output impedances can be dynamically calibrated to compensate for variations in voltage and temperature.

The ODT feature can be disabled by the controller. When ODT is enabled by the controller, the SSTL I/O automatically enables its internal ODT circuitry when in Input mode and disables this circuitry when in Output mode. The initial programming and subsequent calibration of the ODT and output impedance can be triggered to calibrate the ODT and output impedance values at the I/Os based on the desired impedance value when compared to a precision external resistor.

18.3.2 Programming Model

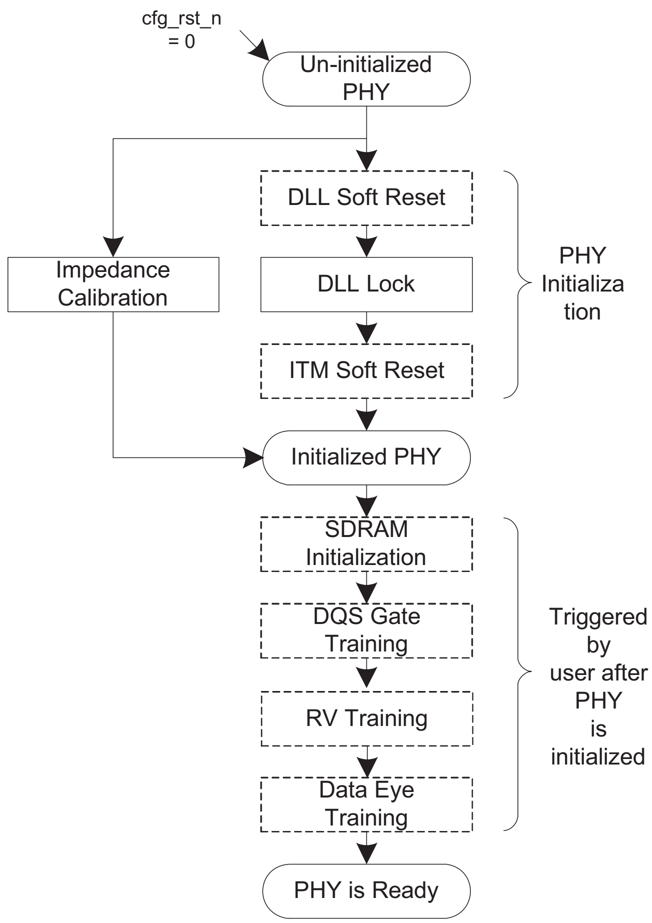
18.3.2.1 Initialization

The initialization sequence has two phases. The first phase happens automatically at reset and is as follows:

1. Before and during configuration reset (i.e. if cfg_rst_n is asserted), the PHY is un-initialized and remains in this state until the reset is de-asserted.
2. At reset de-assertion, the PHY moves into the DLL initialization (lock) phase. This phase may be bypassed at any time by writing a '1' to the DLL initialization bypass register bit (DDR3PHY_PIR.LOCKBYP).
3. In parallel to DLL initialization, the impedance calibration phase also starts at reset de-assertion. This phase can also be bypassed by writing a '1' to the impedance calibration bypass register bit (DDR3PHY_PIR.ZCALBYP).
4. If the PHY initialization sequence was triggered by the user, a soft reset may optionally be selected to be issued to the ITMs. Initialization that is automatically triggered on reset does not issue a soft reset to the ITMs because the components have already been reset by the main reset.
5. Once the DLL initialization and impedance calibration phases are done and after the ITMs are reset, the PHY is initialized. Note that if these phases were bypassed, it is up to the user to perform them using software, or to trigger them at a later time before the PHY can be used.

The following figure shows a high-level illustration of the PHY initialization sequence.

Figure 18-1. DDR3PHY Initialization Flow



The above figure shows the timing diagram for the first phase of PHY initialization which happens automatically after reset is de-asserted.

The DLL soft reset and ITM soft reset are executed only when the first phase is triggered by the user through the PHY Initialization register (DDR3PHY_PIR).

The second initialization phase starts after the PHY is initialized. Each step of this phase is triggered by the user or memory controller:

1. The user or memory controller performs SDRAM initialization sequence. The DRAM mode registers and necessary timing parameters must first be programmed before triggering SDRAM initialization.
2. After the SDRAM is initialized, the user or memory controller performs DQS gate training ("Built-in DQS Gate Training"). The SDRAM must be initialized before triggering DQS gate training.
3. The user or memory controller performs read data valid training.
4. The user or memory controller performs read data eye training.
5. The PHY is now ready for SDRAM read/write accesses.

18.3.2.2 SDRAM Initialization

Important: The PHY registers, SDRAM mode registers, and equivalent register fields inside the memory controller (UDDRC) must be uniformly programmed. Mismatches between the register fields can cause transaction failures. Verify all register fields are consistently programmed before starting any SDRAM transactions.

Prior to normal operation, DDR SDRAMs must be initialized. The PHY has a built-in SDRAM initialization routine that may be triggered by software or memory controller by writing to DDR3PHY_PIR. The initialization routine built into the PHY is generic and does not require any knowledge of the type or configuration of external SDRAMs to be properly executed. The routine is designed with the relevant JEDEC specifications for the fastest and slowest SDRAMs supported by the PHY to result in a universal initialization sequence. This generic sequence is applied to DDR3, DDR2, LPDDR3 and LPDDR2 SDRAMs during the SDRAM initialization sequence. Refer to "SDRAM Initialization Sequence" in the section "Universal DDR Memory Controller (UDDRC)".

18.3.3 PHY Registers

The PHY has registers that are used to configure, control or provide status of certain features of the PHY. The user accesses these registers using the configuration port write and read commands.

All registers are 32 bits wide. For reserved register addresses or reserved register bits, writes have no effect and reads return 0. The following sections describe the PHY registers in detail. Unless specified otherwise, control register bits are assumed active-high.

18.4 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	DDR3PHY_PIR	31:24	INITBYP	ZCALBYP	LOCKBYP	CLRSR				
		23:16						CTLDINIT	DLLBYP	ICPC
		15:8								RVTRN
		7:0	QSTRN	DRAMINIT	DRAMRST	ITMSRST	ZCAL	DLLLOCK	DLLSRST	INIT
0x08	DDR3PHY_PGCR	31:24	LBMODE	LBGDQS	LBDQSS	RFSHDT[3:0]				PDDISDX
		23:16	ZCKSEL[1:0]		RANKEN[3:0]				IODDRM[1:0]	
		15:8	IOLB	CKINV	CKDV[1:0]		CKEN[2:0]			DTOSEL[3]
		7:0	DTOSEL[2:0]			DFTLMT[1:0]		DFTCMP	DQSCFG	ITMDMD
0x0C	DDR3PHY_PGSR	31:24								
		23:16								
		15:8							RVEIRR	RVERR
		7:0	DFTERR	DTIERR	DTERR	DTDONE	DIDONE	ZCDONE	DLDONE	IDONE
0x10	DDR3PHY_DLLGCR	31:24			LOCKDET	FDTRM_SL[1:0]		SBIAS[5:3]		
		23:16	BPS200	SBIAS[2:0]			MBIAS[7:4]			
		15:8	MBIAS[3:0]				TESTSW	ATC[1:0]		DTC[2]
		7:0	DTC[1:0]		TESTEN	IPUMP[2:0]			DRES[1:0]	
0x14	DDR3PHY_ACDLLCR	31:24	DLLDIS	DLLSRST						
		23:16						ATESTEN		
		15:8					MFWDLY[2:0]			MFBDLY[2]
		7:0	MFBDLY[1:0]							
0x18	DDR3PHY_PTR0	31:24								
		23:16			tITMSRST[3:0]				tDLLLOCK[11:10]	
		15:8	tDLLLOCK[9:2]							
		7:0	tDLLLOCK[1:0]		tDLLSRST[5:0]					
0x1C	DDR3PHY_PTR1	31:24						tDINIT1[7:5]		
		23:16	tDINIT1[4:0]					tDINIT0[18:16]		
		15:8	tDINIT0[15:8]							
		7:0	tDINIT0[7:0]							
0x20	DDR3PHY_PTR2	31:24						tDINIT3[9:7]		
		23:16	tDINIT3[6:0]							tDINIT2[16]
		15:8	tDINIT2[15:8]							
		7:0	tDINIT2[7:0]							
0x24	DDR3PHY_ACIOCR	31:24	ACSR[1:0]		RSTIOM	RSTPDR	RSTPDD	RSTODT	RANKPDR[3:2]	
		23:16	RANKPDR[1:0]		CSPDD[3:0]				RANKODT[3:2]	
		15:8	RANKODT[1:0]		CKPDR[2:0]			CKPDD[2:0]		
		7:0	CKODT[2:0]			ACPDR	ACPDD	ACODT	ACOE	ACIOM
0x28	DDR3PHY_DXCCR	31:24								
		23:16								AWDT
		15:8	RVSEL	DQSNRST	DXSR[1:0]		DQSNRES[3:0]			
		7:0	DQSRES[3:0]				DXPDR	DXPDD	DXIOM	DXODT
0x2C	DDR3PHY_DSGCR	31:24	CKEOE	RSTOE	ODTOE	CKOE	TPDOE	TPDPD	NL2OE	NL2PD
		23:16	ODTPDD[3:0]				CKEPDD[3:0]			
		15:8				FXDLAT	NOBUB	DQSGE[2:0]		
		7:0	DQSGX[2:0]			LPDLLPD	LPIOPD	ZUEN	BDISEN	PUREN
0x30	DDR3PHY_DCR	31:24	TPD	RDIMM	UDIMM	DDR2T	NOSRA			
		23:16								
		15:8							DDRTYPE[1:0]	
		7:0	MPRDQ	PDQ[2:0]			DDR8BNK	DDRMD[2:0]		
0x34	DDR3PHY_DTPR0	31:24	tCCD	tRC[5:0]						tRRD[3]
		23:16	tRRD[2:0]			tRAS[4:0]				
		15:8	tRCD[3:0]				tRP[3:0]			
		7:0	tWTR[2:0]			tRTP[2:0]			tMRD[1:0]	

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	DDR3PHY_DTPR1	31:24			tDQSCKmax[2:0]			tDQSCKmin[2:0]		
		23:16	tRFC[7:0]							
		15:8					tRTODT	tMOD[1:0]		tFAW[5]
		7:0	tFAW[4:0]					tRTW	tAOND/ tAOFD[1:0]	
0x3C	DDR3PHY_DTPR2	31:24				tDLLK[9:5]				
		23:16	tDLLK[4:0]					tCKE[3:1]		
		15:8	tCKE[0]	tXP[4:0]					tXS[9:8]	
		7:0	tXS[7:0]							
0x40	DDR3PHY_MR0_DDR3	31:24								
		23:16								
		15:8				PD	WR[2:0]			DR
		7:0	TM	CL3	CL2	CL1	BT	CL0	BL[1:0]	
0x40	DDR3PHY_MR0_DDR2	31:24								
		23:16								
		15:8				PD	WR[2:0]			DR
		7:0	TM	CL[2:0]			BT	BL[2:0]		
0x40	DDR3PHY_MR0_DDR	31:24								
		23:16								
		15:8								DR
		7:0	TM	CL[2:0]			BT	BL[2:0]		
0x44	DDR3PHY_MR1_DDR3	31:24								
		23:16								
		15:8				QOFF	TDQS		RTT2	
		7:0	LEVEL	RTT1	DIC1	AL[1:0]		RTT0	DIC0	DE
0x44	DDR3PHY_MR1_DDR2	31:24								
		23:16								
		15:8				QOFF	RDQS	DQS	OCD[2:1]	
		7:0	OCD[0]	RTT1	AL[2:0]			RTT0	DIC	DE
0x44	DDR3PHY_MR1_DDR	31:24								
		23:16								
		15:8								
		7:0							DS	DE
0x44	DDR3PHY_MR1_LPDDR2	31:24								
		23:16								
		15:8								
		7:0	nWR[2:0]			WC	BT	BL[2:0]		
0x44	DDR3PHY_MR1_LPDDR3	31:24								
		23:16								
		15:8								
		7:0	nWR[2:0]					BL[2:0]		
0x48	DDR3PHY_MR2_DDR3	31:24								
		23:16								
		15:8	RSVD[4:0]					RTTWR[1:0]		
		7:0	SRT	ASR	CWL[2:0]			PASR[2:0]		
0x48	DDR3PHY_MR2_DDR2	31:24								
		23:16								
		15:8	RSVD[4:0]							
		7:0	SRF				DCC	PASR[2:0]		
0x48	DDR3PHY_MR2_LPDDR2	31:24								
		23:16								
		15:8								
		7:0	RSVD[3:0]				RL/WL[3:0]			
0x48	DDR3PHY_MR2_LPDDR3	31:24								
		23:16								
		15:8								
		7:0	RSVD[3:0]				RL/WL[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	DDR3PHY_MR3_DDR3	31:24								
		23:16								
		15:8	RSVD[12:5]							
		7:0	RSVD[4:0]					MPR	MPRLOC[1:0]	
0x4C	DDR3PHY_MR3_EMR3_DDR2	31:24								
		23:16								
		15:8	RSVD[15:8]							
		7:0	RSVD[7:0]							
0x4C	DDR3PHY_MR3_LPDDR2	31:24								
		23:16								
		15:8								
		7:0	RSVD[3:0]				DS[3:0]			
0x4C	DDR3PHY_MR_LPDDR3	31:24								
		23:16								
		15:8								
		7:0	RSVD[3:0]				PDCTL[1:0]		DQODT[1:0]	
0x50	DDR3PHY_ODTCR	31:24	WRODT3[3:0]				WRODT2[3:0]			
		23:16	WRODT1[3:0]				WRODT0[3:0]			
		15:8	RDODT3[3:0]				RDODT2[3:0]			
		7:0	RDODT1[3:0]				RDODT0[3:0]			
0x54	DDR3PHY_DTAR	31:24	DTMPR	DTBANK[2:0]			DTROW[15:12]			
		23:16	DTROW[11:4]							
		15:8	DTROW[3:0]				DTCOL[11:8]			
		7:0	DTCOL[7:0]							
0x58	DDR3PHY_DTDR0	31:24	DTBYTE3[7:0]							
		23:16	DTBYTE2[7:0]							
		15:8	DTBYTE1[7:0]							
		7:0								DTBYTE0
0x5C	DDR3PHY_DTDR1	31:24	DTBYTE7[7:0]							
		23:16	DTBYTE6[7:0]							
		15:8	DTBYTE5[7:0]							
		7:0	DTBYTE4[7:0]							
0x60 ... 0xBF	Reserved									
0xC0	DDR3PHY_DCUAR	31:24								
		23:16								
		15:8					ATYPE	INCA	CSEL[1:0]	
		7:0	CSADDR[3:0]				CWADDR[3:0]			
0xC4	DDR3PHY_DCUDR	31:24	CDATA[31:24]							
		23:16	CDATA[23:16]							
		15:8	CDATA[15:8]							
		7:0	CDATA[7:0]							
0xC8	DDR3PHY_DCURR	31:24								
		23:16	XCEN	RCEN	SCOF	SONF	NFAIL[7:4]			
		15:8	NFAIL[3:0]				EADDR[3:0]			
		7:0	SADDR[3:0]				DINST[3:0]			
0xCC	DDR3PHY_DCULR	31:24	XLEADDR[3:0]							
		23:16							IDA	LINF
		15:8	LCNT[7:0]							
		7:0	LEADDR[3:0]				LSADDR[3:0]			
0xD0	DDR3PHY_DCUGCR	31:24								
		23:16								
		15:8	RCSW[15:8]							
		7:0	RCSW[7:0]							
0xD4	DDR3PHY_DCUTPR	31:24	tDCUT3[7:0]							
		23:16	tDCUT2[7:0]							
		15:8	tDCUT1[7:0]							
		7:0	tDCUT0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xD8	DDR3PHY_DCUSR0	31:24								
		23:16								
		15:8								
		7:0						CFULL	CFAIL	RDONE
0xDC	DDR3PHY_DCUSR1	31:24	LPCNT[7:0]							
		23:16	FLCND[7:0]							
		15:8	RDCNT[15:8]							
		7:0	RDCNT[7:0]							
0xE0 ... 0xFF	Reserved									
0x0100	DDR3PHY_BISTRR	31:24							BCKSEL[2:1]	
		23:16	BCKSEL[0]			BDXSEL[1:0]		BDPAT[1:0]		BDMEN
		15:8	BACEN	BDXEN	BSONF	NFAIL[7:3]				
		7:0	NFAIL[2:0]			BINF	BMODE	BINST[2:0]		
0x0104	DDR3PHY_BISTMSKR0	31:24				ODTMSK				CSMSK
		23:16				CKEMSK	WEMSK	BAMSK[2:0]		
		15:8	AMSK[15:8]							
		7:0	AMSK[7:0]							
0x0108	DDR3PHY_BISTMSKR1	31:24								
		23:16					CASMSK	RASMSK	DMMSK[1:0]	
		15:8								
		7:0							DQMSK[1:0]	
0x010C	DDR3PHY_BISTWCR	31:24								
		23:16								
		15:8	BWCNT[15:8]							
		7:0	BWCNT[7:0]							
0x0110	DDR3PHY_BISTLSR	31:24	SEED[31:24]							
		23:16	SEED[23:16]							
		15:8	SEED[15:8]							
		7:0	SEED[7:0]							
0x0114	DDR3PHY_BISTAR0	31:24		BBANK[2:0]			BROW[15:12]			
		23:16	BROW[11:4]							
		15:8	BROW[3:0]				BCOL[11:8]			
		7:0	BCOL[7:0]							
0x0118	DDR3PHY_BISTAR1	31:24								
		23:16								
		15:8	BAINC[11:4]							
		7:0	BAINC[3:0]				BMRANK[1:0]		BRANK[1:0]	
0x011C	DDR3PHY_BISTAR2	31:24		BMBANK[2:0]			BMROW[15:12]			
		23:16	BMROW[11:4]							
		15:8	BMROW[3:0]				BMCOL[11:8]			
		7:0	BMCOL[7:0]							
0x0120	DDR3PHY_BISTUDPR	31:24	BUDP1[15:8]							
		23:16	BUDP1[7:0]							
		15:8	BUDP0[15:8]							
		7:0	BUDP0[7:0]							
0x0124	DDR3PHY_BISTGSR	31:24	CASBER[1:0]		RASBER[1:0]		DMBER[3:0]			
		23:16								
		15:8								
		7:0						BDXERR	BACERR	BDONE
0x0128	DDR3PHY_BISTWER	31:24	DXWER[15:8]							
		23:16	DXWER[7:0]							
		15:8	ACWER[15:8]							
		7:0	ACWER[7:0]							
0x012C	DDR3PHY_BISTBER0	31:24	ABER[31:24]							
		23:16	ABER[23:16]							
		15:8	ABER[15:8]							
		7:0	ABER[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0130	DDR3PHY_BISTBER1	31:24							ODTBER[1:0]	
		23:16							CSBER[1:0]	
		15:8							CKEBER[1:0]	
		7:0	WEBER[1:0]		BABER[5:0]					
0x0134	DDR3PHY_BISTBER2	31:24	DQBER[31:24]							
		23:16	DQBER[23:16]							
		15:8	DQBER[15:8]							
		7:0	DQBER[7:0]							
0x0138	DDR3PHY_BISTWCSR	31:24	DXWCNT[15:8]							
		23:16	DXWCNT[7:0]							
		15:8	ACWCNT[15:8]							
		7:0	ACWCNT[7:0]							
0x013C	DDR3PHY_BISTFWR0	31:24				ODTWEBS				CSWEBS
		23:16				CKEWEBS	WEWEBS	BAWEBS[2:0]		
		15:8	AWEBS[15:8]							
		7:0	AWEBS[7:0]							
0x0140	DDR3PHY_BISTFWR1	31:24								
		23:16					CASWEBS	RASWEBS	DMWEBS[1:0]	
		15:8	DQWEBS[15:8]							
		7:0	DQWEBS[7:0]							
0x0144 ... 0x017F	Reserved									
0x0180	DDR3PHY_ZQ0CR0	31:24	ZQPD	ZCAL	ZCALBYP	ZDEN	ZDATA[27:24]			
		23:16	ZDATA[23:16]							
		15:8	ZDATA[15:8]							
		7:0	ZDATA[7:0]							
0x0184	DDR3PHY_ZQ0CR1	31:24								
		23:16								
		15:8								
		7:0	ZPROG[7:0]							
0x0188	DDR3PHY_ZQ0SR0	31:24	ZDONE	ZERR			ZCTRL[27:24]			
		23:16	ZCTRL[23:16]							
		15:8	ZCTRL[15:8]							
		7:0	ZCTRL[7:0]							
0x018C	DDR3PHY_ZQ0SR1	31:24								
		23:16								
		15:8								
		7:0	OPU[1:0]		OPD[1:0]		ZPU[1:0]		ZPD[1:0]	
0x0190 ... 0x01BF	Reserved									
0x01C0	DDR3PHY_DX0GCR	31:24								
		23:16								R0RVSL[2]
		15:8	R0RVSL[1:0]		RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1]
		7:0	DSEN[0]	DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
0x01C4	DDR3PHY_DX0GSR0	31:24								
		23:16								
		15:8	DTPASS[2:0]							DTIERR
		7:0				DTERR				DTDONE
0x01C8	DDR3PHY_DX0GSR1	31:24								
		23:16			RVPASS[1:0]					RVIERR
		15:8				RVERR				
		7:0			DQSDFT[1:0]					DFTERR
0x01CC	DDR3PHY_DX0DLLCR	31:24	DLLDIS	DLLSRST						
		23:16					SDLBMODE	ATESTEN	SDPHASE[3:2]	
		15:8	SDPHASE[1:0]		SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2]
		7:0	MFBDLY[1:0]		SFWDLY[2:0]			SFBDLY[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01D0	DDR3PHY_DX0DQTR	31:24								
		23:16								
		15:8								
		7:0	DQDLY1[3:0]				DQDLY0[3:0]			
0x01D4	DDR3PHY_DX0DQSTR	31:24							DQSNDLY[2:1]	
		23:16	DQSNDLY[0]	DQSDLY[2:0]						
		15:8			R0DGPS[1:0]					
		7:0						R0DGSL[2:0]		
0x01D8 ... 0x01FF	Reserved									
0x0200	DDR3PHY_DX1GCR	31:24								
		23:16								R0RVSL[2]
		15:8	R0RVSL[1:0]		RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1]
		7:0	DSEN[0]	DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
0x0204	DDR3PHY_DX1GSR0	31:24								
		23:16								
		15:8	DTPASS[2:0]							DTIERR
		7:0				DTERR				DTDONE
0x0208	DDR3PHY_DX1GSR1	31:24								
		23:16			RVPASS[1:0]					RVIERR
		15:8				RVERR				
		7:0			DQSDFT[1:0]					DFTERR
0x020C	DDR3PHY_DX1DLLCR	31:24	DLLDIS	DLLSRST						
		23:16					SDLBMODE	ATESTEN	SDPHASE[3:2]	
		15:8	SDPHASE[1:0]		SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2]
		7:0	MFBDLY[1:0]		SFWDLY[2:0]			SFBDLY[2:0]		
0x0210	DDR3PHY_DX1DQTR	31:24								
		23:16								
		15:8								
		7:0	DQDLY1[3:0]				DQDLY0[3:0]			
0x0214	DDR3PHY_DX1DQSTR	31:24							DQSNDLY[2:1]	
		23:16	DQSNDLY[0]	DQSDLY[2:0]						
		15:8			R0DGPS[1:0]					
		7:0						R0DGSL[2:0]		

18.4.1 DDR3PHY PHY Initialization Register

Name: DDR3PHY_PIR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

PHY Initialization Register is used to configure and control the initialization of the PHY. This includes the triggering of certain initialization routines, as well as the reset of the PHY and/or the DLLs used in the PHY. Any configuration register write that sets the PIR[INIT] register bit will trigger initialization as selected by the other PIR register bits. The completion status of this initialization can be checked by polling the PHY General Status Register (PGSR). Note that PGSR0[IDONE] is not cleared immediately after PIR[INIT] is set, and therefore software must wait a minimum of 10 configuration clock cycles from when PIR[INIT] is set to when it starts polling for PGSR[IDONE].

Bit	31	30	29	28	27	26	25	24
	INITBYP	ZCALBYP	LOCKBYP	CLRSR				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
						CTLDINIT	DLLBYP	ICPC
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
								RVTRN
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	QSTRN	DRAMINIT	DRAMRST	ITMSRST	ZCAL	DLLLOCK	DLLSRST	INIT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – INITBYP Initialization Bypass
Bypasses or stops, if set, all initialization routines currently running, including PHY initialization, DRAM initialization, and PHY training. Initialization may be triggered manually using INIT and the other relevant bits of the PIR register. This bit is self-clearing.

Bit 30 – ZCALBYP Impedance Calibration Bypass
Bypasses or stops, if set, impedance calibration of all ZQ control blocks that automatically triggers after reset. Impedance calibration may be triggered manually using INIT and ZCAL bits of the PIR register. This bit is self-clearing.

Bit 29 – LOCKBYP DLL Lock Bypass
Bypasses or stops, if set, the waiting of DLLs to lock. DLL lock wait is automatically triggers after reset. DLL lock wait may be triggered manually using INIT and DLLLOCK bits of the PIR register. This bit is self-clearing.

Bit 28 – CLRSR Clear Status Registers
Writing 1 to this bit does the following:
- Auto-clears itself (PIR.CLRSR). This means reads to PIR.CLRSR will return 0.
- Clears the DXnGSR0 bits for DTDONE, DTERR, DTIERR
- Clears the PGSR.DFTERR and DXnGSR1.DFTERR bits

This bit is primarily for debug purposes and is typically not needed during normal functional operation. It can be used when PGSR.IDONE=1, to manually clear the PGSR status bits, however, the PGSR status bits (except for the DFTRR and TQ bits) are automatically cleared by starting a new init process.
The bit can also be used to manually clear the DXnGSR status bits, however, starting a new data training process automatically clears the DXnGSR status bits.

Bit 18 – CTLDINIT Controller DRAM Initialization
Indicates if set that DRAM initialization will be performed by the controller. Otherwise if not set it indicates that DRAM initialization will be performed using the built-in initialization sequence or using software through the configuration port.

Bit 17 – DLLBYP DLL Bypass
A setting of 1 on this bit will put all PHY DLLs in Bypass mode. A bypassed DLL is also powered down (disabled).
Note: Ensure that the minimum requirements for DLL bypass and DLL reset are observed while asserting this bit.

Bit 16 – ICPC Initialization Complete Pin Configuration
Specifies how the DFI 2.1 initialization complete output pin should be used to indicate the status of initialization.

Value	Description
0	Asserted after PHY initialization (DLL locking and impedance calibration) is complete.
1	Asserted after PHY initialization is complete and the triggered the PUBL initialization (DRAM initialization, data training, or initialization trigger with no selected initialization) is complete.

Bit 8 – RVTRN RV Training
Read DQS Gate Training (QSTRN) and RV Training (RVTRN) should normally be run together. It is expected RVTRN is normally be set whenever PIR.QSTRN is set.
If RVTRN=1 and PIR.QSTRN=0, when PIR.INIT=1 (triggering the init process) the read DQS gate training algorithm will still be run (as if PIR.QSTRN were actually set to 1).
If it is necessary to run only RV training stand-alone, with no read DQS gate training, you can prevent the read DQS gate training from running by setting PGCR.LBGDQS=1 (see 18.4.2. DDR3PHY_PGCR).
Note: RV Training cannot use Multi-Purpose Register (MPR) and must use the user data programmed in DTDR0-1.

Bit 7 – QSTRN Read DQS Training
Executes a PUBL training routine to determine the optimum position of the read data DQS strobe for maximum system timing margins.

Bit 6 – DRAMINIT DRAM Initialization
Executes the DRAM initialization sequence.

Bit 5 – DRAMRST DRAM Reset (DDR3 Only)
Issues a reset to the DRAM (by driving the DRAM reset pin low) and wait 200 µs. This can be triggered in isolation or with the full DRAM initialization (DRAMINIT). For the latter case, the reset is issued and 200 µs is waited before starting the full initialization sequence.

Bit 4 – ITMSRST Interface Timing Module Soft Reset
Soft resets the interface timing modules for the data and data strobes, i.e., it asserts the ITM soft reset (srstb) signal.

Bit 3 – ZCAL Impedance Calibrate
Performs PHY impedance calibration.

Bit 2 – DLLLOCK DLL Lock
Waits for the PHY DLLs to lock.

Bit 1 – DLLSRST DLL Soft Rest
Soft resets all PHY DLLs by driving the DLL soft reset pin.
- Requires that ctl_clk be toggling for the DLL soft reset signal to be output from the DLL. If ctl_clk (AC DLL's cclk_0) is not guaranteed to be toggling, it is recommended to use the manual DLL soft reset ACDLLCR[DLLSRST], not PIR[DLLSRST].
- Ensure that the minimum requirements for DLL bypass and DLL reset are observed while asserting this bit.

Bit 0 – INIT Initialization Trigger
A write of '1' to this bit triggers the DDR system initialization, including PHY initialization, DRAM initialization, and PHY training. The exact initialization steps to be executes are specified in bits 1 to 6 of this register. A bit setting of

1 means the step will be executed as part of the initialization sequence, while a setting of ‘0’ means the step will be bypassed. The initialization trigger bit is self-clearing.

18.4.2 DDR3PHY PHY General Configuration Register

Name: DDR3PHY_PGCR
Offset: 0x08
Reset: 0x01BC2E04
Property: Read/Write

This register is used for miscellaneous PHY configurations such as PHY type, DQS gating configuration, and DQS drift compensation.

Bit	31	30	29	28	27	26	25	24
	LBMODE	LBGDQS	LBDQSS	RFSHDT[3:0]				PDDISDX
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	ZCKSEL[1:0]		RANKEN[3:0]				IODDRM[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	1	1	1	0	0

Bit	15	14	13	12	11	10	9	8
	IOLB	CKINV	CKDV[1:0]		CKEN[2:0]			DTOSEL[3]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	1	1	1	0

Bit	7	6	5	4	3	2	1	0
	DTOSEL[2:0]			DFTLMT[1:0]		DFTCMP	DQSCFG	ITMDMD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit 31 – LBMODE Loopback Mode
Indicates if set that the PHY/PUBL is in Loopback mode.

Bit 30 – LBGDQS Loopback DQS Gating
Selects the DQS gating mode that should be used when the PHY is in Loopback mode, including BIST Loopback mode.
Note: When PGCR.LBGDQS=0, PIR.QSTRN should not be run prior to running BIST DATX8 loopback.

Value	Description
0	DQS gate training will be triggered on the PUBL.
1	DQS gate is set manually using software.

Bit 29 – LBDQSS Loopback DQS Shift
Selects how the read DQS is shifted during loopback to ensure that the read DQS is centered into the read data eye.

Value	Description
0	PUBL sets the read DQS delay to 0, DQS is already shifted 90 degrees by write path.
1	The read DQS shift is set manually through software.

Bits 28:25 – RFSHDT[3:0] Refresh During Training
A non-zero value specifies that a burst of refreshes equal to the number specified in this field should be sent to the SDRAM after training each rank except the last rank.

Bit 24 – PDDISDX Power Down Disabled Byte
Indicates if set that the DLL and I/Os of a disabled byte should be powered down.

Bits 23:22 – ZCKSEL[1:0] Impedance Clock Divider Select
Selects the divide ratio for the clock used by the impedance control logic. The source clock for the divider is the configuration port clock signal (cfg_clk or pclk), depending on which configuration port type used.

Value	Description
00	Divide by 2
01	Divide by 8
10	Divide by 32
11	Divide by 64

Bits 21:18 – RANKEN[3:0] Rank Enable
Specifies the ranks that are enabled for data training. Bit 0 controls rank 0, bit 1 controls rank 1, bit 2 controls rank 2, and bit 3 controls rank 3. Setting the bit to ‘1’ enables the rank, and setting it to ‘0’ disables the rank.

Bits 17:16 – IODDRM[1:0] I/O DDR Mode (D3F I/O Only)
Selects the DDR mode for the I/Os.

Bit 15 – IOLB I/O Loop-Back Select
Selects where, inside the I/O, the signal loopback happens. Not applicable to D3A I/Os.

Value	Description
0	Loopback is after output buffer, output enable must be asserted.
1	Loopback is before output buffer, output enable is don’t care.

Bit 14 – CKINV CK Invert
Specifies if set that CK/CK# should be inverted. Otherwise CK/CK# toggles with normal polarity.

Bits 13:12 – CKDV[1:0] CK Disable Value
Specifies the static value that should be driven on CK/CK# pair(s) when the pair(s) is disabled. CKDV[0] specifies the value for CK and CKDV[1] specifies the value for CK#.

Bits 11:9 – CKEN[2:0] CK Enable
Controls whether the CK going to the SDRAM is enabled (toggling) or disabled (static value defined by CKDV). One bit for each of the three CK pairs.

Bits 8:5 – DTOSEL[3:0] Digital Test Output Select
Selects the PHY digital test output that should be driven onto PHY digital test output (phy_dto) pin.

Value	Description
0000	DATX8 0 DLL digital test output
0001	DATX8 1 DLL digital test output
0010	DATX8 2 DLL digital test output
0011	DATX8 3 DLL digital test output
0100	DATX8 4 DLL digital test output
0101	DATX8 5 DLL digital test output
0110	DATX8 6 DLL digital test output
0111	DATX8 7 DLL digital test output
1000	DATX8 8 DLL digital test output
1001	AC DLL digital test output
1010 – 0111	Reserved

Bits 4:3 – DFTLMT[1:0] DQS Drift Limit
Specifies the expected limit of drift on read data strobes. A drift of this value or greater is reported as a drift error through the host port error flag.
Note: Although reported through the error flag, this is not an error requiring any action. It is simply an indicator that the drift is greater than expected.

Value	Description
00	No limit (no error reported)
01	90° drift
10	180° drift
11	270° or more drift

Bit 2 – DFTCMP DQS Drift Compensation
Enables or disables DQS drift compensation. By default, drift compensation is enabled. Drift compensation is not supported under any of the following situations:
- LPDDR2/3 (DCR.DDRMD set to LPDDR2)
- Burst length 2 (MR0.BL set to burst length of 2)
- Read DQS gating using passive windowing (PGCR.DQSCFG set to passive windowing)

Drift compensation must be set to disabled if any of the above are set.

Value	Description
0	Disables data strobe drift compensation.
1	Enables data strobe drift compensation.

Bit 1 – DQSCFG DQS Gating Configuration
Selects one of the two DQS gating schemes.
Note: Passive windowing must be used for LPDDR2/3.

Value	Description
0	DQS gating is shut off using the rising edge of DQS_b (Active Windowing mode).
1	DQS gating blankets the whole burst (Passive Windowing mode).

Bit 0 – ITMDMD ITM DDR Mode
Selects whether ITMS uses DQS and DQS# or only DQS.
Note: The correct setting when using DDR is 1 because neither uses DQS#.

Value	Description
0	ITMS uses DQS and DQS#.
1	ITMS uses DQS only.

18.4.3 DDR3PHY PHY General Status Register

Name: DDR3PHY_PGSR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

This is a general status register for the PHY. It indicates, among other things, whether data training is done.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RVEIRR	RVERR
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	DFTERR	DTIERR	DTERR	DTDONE	DIDONE	ZCDONE	DLDONE	IDONE
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RVEIRR Read Valid Training Intermittent Error
If set, indicates that there was an intermittent error during read valid training, such as a pass was followed by a fail then followed by another pass.

Bit 8 – RVERR Read Valid Training Error
If set, indicates that a valid read valid placement could not be found during read valid training.

Bit 7 – DFTERR DQS Drift Error
If set, indicates that at least one of the read data strobes has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR).

Bit 6 – DTIERR DQS Gate Training Intermittent Error
If set, indicates that there was an intermittent error during DQS gate training, such as a pass was followed by a fail then followed by another pass.

Bit 5 – DTERR DQS GateTraining Error
If set, indicates that a valid DQS gating window could not be found during DQS gate training.

Bit 4 – DTDONE Data Training Done
Indicates, if set, that the PHY has finished doing data training.

Bit 3 – DIDONE DRAM Initialization Done
Indicates if set that DRAM initialization has completed.

Bit 2 – ZCDONE Impedance Calibration Done
Indicates if set that impedance calibration has completed.

Bit 1 – DLDONE DLL Lock Done
Indicates if set that DLL locking has completed.

Bit 0 – IDONE Initialization Done
Indicates if set that the DDR system initialization has completed. This bit is set after all the selected initialization routines in PIR register have completed.

18.4.4 DDR3PHY DLL General Control Register

Name: DDR3PHY_DLLGCR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

The DLLGCR register provides miscellaneous global configuration and trimming of all the DLLs used in the AC and DATX8 macros, including DLL test modes. Some fields of this register have different default values depending on the technology process. Similarly, some register fields are reserved for other processes.

Bit	31	30	29	28	27	26	25	24
			LOCKDET	FDTRM_SL[1:0]		SBIAS[5:3]		
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BPS200	SBIAS[2:0]			MBIAS[7:4]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MBIAS[3:0]				TESTSW	ATC[1:0]		DTC[2]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DTC[1:0]		TESTEN	IPUMP[2:0]			DRES[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – LOCKDET Host Lock Detector Enable
This field is only valid for a few processes. It is reserved for all other processes.

Bits 28:27 – FDTRM_SL[1:0] Client Bypass Fixed Delay Trim

Value	Description
`00`	Nominal delay
`01`	Nominal delay – 10%
`10`	Nominal delay + 10%
`11`	Nominal delay + 20%

Bits 26:24 – SBIAS[5:3] Client Bias Trim
Used to trim the bias for the client DLL.

Bit 23 – BPS200 Bypass Mode Frequency Range

Value	Description
`0`	0 to 100 MHz
`1`	0 to 200 MHz

Bits 22:20 – SBIAS[2:0] Client Bias Trim
Used to trim the bias for the client DLL.

Bits 19:12 – MBIAS[7:0] Host Bias Trim
Used to trim the bias for the host DLL.

Bit 11 – TESTSW Test Switch
Selects the test signals of either the host DLL, set to 0, or the client DLL, set to 1.

Bits 10:9 – ATC[1:0] Analog Test Control
Selects the analog signal to be output on the DLL analog test output (test_out_a) when TESTEN is high (Output is Vss when TESTEN is low). The test output either comes from the host DLL or the client DLL, depending on the setting of the test switch (TESTSW). Both host DLL and client DLL output similar analog test signals.

Value	Description
00	Replica bias output for PMOS (Vbp)
01	Replica bias output for NMOS (Vbn)
10	Filter output (Vc)
11	Vdd

Bits 8:6 – DTC[2:0] Digital Test Control
Selects the digital signal to be output on the DLL digital test output (test_out_d[1]) when TESTEN is high (Output is '0' when TESTEN is low).

Value	Description
	For Host DLL (TESTSW = 0):
000	0 output clock (clk_0)
001	90 output clock (clk_90)
010	180 output clock (clk_180)
011	270 output clock (clk_270)
100	360 internal clock (clk_360_int)
101	Speed-up pulse (spdup)
110	Slow-down pulse (slwdn)
111	0 MCTL logic clock (cclk_0)
	For Client DLL (TESTSW = 1):
000	Input DQS strobe (dqs)
001	Input clock reference (clk_90_in)
010	Internal feedback clock (clk_0_out)
011	90 output DQS_b strobe (dqsb_90)
100	90 output DQS strobe (dqs_90)
111	Auto-lock enable signal

Bit 5 – TESTEN Test Enable
Enables digital and analog test outputs selected by DTC and ATC respectively.

Bits 4:2 – IPUMP[2:0] Charge Pump Current Trim
Used to trim charge pump current.

Value	Description
000	Maximum current
100	Minimum current

Bits 1:0 – DRES[1:0] Delta Resistor Trim
Used to trim reference current versus resistor value variation.
Note: For a few processes this field is reserved and is not used by the DLL.

Value	Description
00	Rnom
01	Rnom - 20%
1x	Rnom + 20%

18.4.5 DDR3PHY AC DLL Control Register

Name: DDR3PHY_ACDLLCR
Offset: 0x14
Reset: 0x40000000
Property: Read/Write

AC DLL Control Register is used to control the DLL on the address/command lane.

Bit	31	30	29	28	27	26	25	24
	DLLDIS	DLLSRST						
Access	R/W	R/W						
Reset	0	1						

Bit	23	22	21	20	19	18	17	16
						ATESTEN		
Access						R/W		
Reset						0		

Bit	15	14	13	12	11	10	9	8
					MFWDLY[2:0]			MFBDLY[2]
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MFBDLY[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bit 31 – DLLDIS DLL Disable
A disabled DLL is bypassed. Default ('0') is DLL enabled.

Bit 30 – DLLSRST DLL Soft Rest
Soft resets the AC DLL by driving the DLL soft reset pin.

Bit 18 – ATESTEN Analog Test Enable
Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.

Bits 11:9 – MFWDLY[2:0] Host Feed-Forward Delay Trim
Used to trim the delay in the host DLL feedforward path.

Value	Description
000	Minimum delay
111	Maximum delay

Bits 8:6 – MFBDLY[2:0] Host Feed-Back Delay Trim
Used to trim the delay in the host DLL feedback path.

Value	Description
000	Minimum delay
111	Maximum delay

18.4.6 DDR3PHY PHY Timing Register 0

Name: DDR3PHY_PTR0
Offset: 0x18
Reset: 0x0022AF9B
Property: Read/Write

The PHY Timing Register is used to program different timing parameters used by the PUBL logic.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			tITMSRST[3:0]				tDLLLOCK[11:10]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	tDLLLOCK[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	tDLLLOCK[1:0]		tDLLSRST[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	1	0	1	1

Bits 21:18 – tITMSRST[3:0] ITM Soft Reset Time
Number of configuration clock cycles that the ITM soft reset pin must remain asserted when the soft reset is applied to the ITMs. This must correspond to a value that is equal to or more than 8 controller clock cycles. Default value corresponds to 8 controller clock cycles.

Bits 17:6 – tDLLLOCK[11:0] DLL Lock Time
Number of configuration clock cycles for the DLL to stabilize and lock, i.e. number of clock cycles from when the DLL reset pin is de-asserted to when the DLL has locked and is ready for use. Default value corresponds to 5.12 μs at 533 MHz.

Bits 5:0 – tDLLSRST[5:0] DLL Soft Reset Time
Number of configuration clock cycles that the DLL soft reset pin must remain asserted when the soft reset is triggered through the PHY Initialization Register (PIR). This must correspond to a value that is equal to or more than 50 ns or 8 controller clock cycles, whichever is bigger. Default value corresponds to 50 ns at 533 MHz.

18.4.7 DDR3PHY PHY Timing Register 1

Name: DDR3PHY_PTR1
Offset: 0x1C
Reset: 0x0604111D
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						tDINIT1[7:5]		
Access						R/W	R/W	R/W
Reset						1	1	0

Bit	23	22	21	20	19	18	17	16
	tDINIT1[4:0]					tDINIT0[18:16]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
	tDINIT0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	tDINIT0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	0	1

Bits 26:19 – tDINIT1[7:0] DRAM Initialization Time 1
DRAM initialization time corresponding to the following:
DDR3 = CKE high time to first command (tRFC +10 ns or 5 tCK, whichever value is larger)
DDR2 = CKE high time to first command (400 ns)
DDR = CKE high time to first command (400 ns or 1 tCK)
LPDDR2 = CKE low time with power and clock stable (100 ns)
LPDDR3 = CKE low time with power and clock stable (100 ns)
Default value corresponds to DDR3 360 ns at 533 MHz

Bits 18:0 – tDINIT0[18:0] DRAM Initialization Time 0
DRAM initialization time corresponding to the following:
DDR3 = CKE low time with power and clock stable (500 μs)
DDR2 = CKE low time with power and clock stable (200 μs)
LPDDR2 = CKE high time to first command (200 μs)
LPDDR3 = CKE high time to first command (200 μs)
Default value corresponds to DDR3 500 μs at 533 MHz.

18.4.8 DDR3PHY PHY Timing Register 2

Name: DDR3PHY_PTR2
Offset: 0x20
Reset: 0x042DA072
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						tDINIT3[9:7]		
Access						R/W	R/W	R/W
Reset						1	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	tDINIT3[6:0]							tDINIT2[16]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	1	1	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	tDINIT2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	tDINIT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	0	1	0

Bits 26:17 – tDINIT3[9:0] DRAM Initialization Time 3
DRAM initialization time corresponding to the following:
LPDDR2 = Time from ZQ initialization command to first command (1 μs)
LPDDR3 = Time from ZQ initialization command to first command (1 μs)
Default value corresponds to the LPDDR2/3 1 μs at 533 MHz.

Bits 16:0 – tDINIT2[16:0] DRAM Initialization Time 2
DRAM initialization time corresponding to the following:
DDR3 = Reset low time (200 μs on power-up or 100 ns after power-up)
LPDDR2 = Time from reset command to end of auto initialization (1 μs + 10 μs = 11 μs)
LPDDR3 = Time from reset command to end of auto initialization (11 μs)
Default value corresponds to DDR3 200 μs at 533 MHz.

18.4.9 DDR3PHY AC I/O Configuration Register

Name: DDR3PHY_ACIOCR
Offset: 0x24
Reset: 0x33C03812
Property: Read/Write

The AC I/O Configuration Register is used to control output enables, on-die termination enables, and power-down enables for SSTL I/Os for all address/command signals going to the SDRAM, including the SDRAM clock (CK).

For D3A I/Os, the IOM register bits connect to the I/O IDDQ pins. Furthermore, the I/O power-down register bits (PDD, PDR, and PD) fields have no relevance for D3A I/Os.

Bit	31	30	29	28	27	26	25	24
	ACSR[1:0]		RSTIOM	RSTPDR	RSTPDD	RSTODT	RANKPDR[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1

Bit	23	22	21	20	19	18	17	16
	RANKPDR[1:0]		CSPDD[3:0]				RANKODT[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RANKODT[1:0]		CKPDR[2:0]			CKPDD[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
	CKODT[2:0]			ACPDR	ACPDD	ACODT	ACOE	ACIOM
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	1	0

Bits 31:30 – ACSR[1:0] Address/Command Slew Rate
Selects slew rate of the I/O for all address and command pins.

Bit 29 – RSTIOM SDRAM Reset I/O Mode
Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for SDRAM Reset.

Bit 28 – RSTPDR SDRAM Reset Power Down Receiver
Powers down, when set, the input receiver on the I/O for SDRAM RST# pin.

Bit 27 – RSTPDD SDRAM Reset Power Down Driver
Powers down, when set, the output driver on the I/O for SDRAM RST# pin.

Bit 26 – RSTODT SDRAM Reset On-Die Termination
Enables, when set, the on-die termination on the I/O for SDRAM RST# pin.

Bits 25:22 – RANKPDR[3:0] Rank Power Down Receiver
Powers down, when set, the input receiver on the I/O
CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKPDR[0] controls the power down for CKE[0], ODT[0], and CS#[0], RANKPDR[1] controls the power down for CKE[1], ODT[1], and CS#[1], and so on.

Bits 21:18 – CSPDD[3:0] CS# Power Down Driver
Powers down, when set, the output driver on the I/O for CS#[3:0] pins. PDD[0] controls the power down for CS#[0], PDD[1] controls the power down for CS#[1], and so on. CKE and ODT driver power down is controlled by DSGCR register.

Bits 17:14 – RANKODT[3:0] Rank On-Die Termination
Enables, when set, the on-die termination on the I/O for CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKODT[0] controls the on-die termination for CKE[0], ODT[0], and CS#[0], RANKODT[1] controls the on-die termination for CKE[1], ODT[1], and CS#[1], and so on.

Bits 13:11 – CKPDR[2:0] CK Power Down Receiver
Powers down, when set, the input receiver on the I/O for CK[0], CK[1], and CK[2] pins, respectively.

Bits 10:8 – CKPDD[2:0] CK Power Down Driver
Powers down, when set, the output driver on the I/O for CK[0], CK[1], and CK[2] pins, respectively.

Bits 7:5 – CKODT[2:0] CK On-Die Termination
Enables, when set, the on-die termination on the I/O for CK[0], CK[1], and CK[2] pins, respectively.

Bit 4 – ACPDR AC Power Down Receiver
Powers down, when set, the input receiver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.

Bit 3 – ACPDD AC Power Down Driver
Powers down, when set, the output driver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin.

Bit 2 – ACODT Address/Command On-Die Termination
Enables, when set, the on-die termination on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.

Bit 1 – ACOE Address/Command Output Enable
Enables, when set, the output driver on the I/O for all address and command pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.

Bit 0 – ACIOM Address/Command I/O Mode
Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for all address and command pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.

18.4.10 DDR3PHY DATX8 Common Configuration Register

Name: DDR3PHY_DXCCR
Offset: 0x28
Reset: 0x00000800
Property: Read/Write

The DATX8 Common Configuration Register is used to control features that affect all the byte lanes. These include on-die termination enables and power-down enables for SSTL I/Os for all SDRAM data, data mask, and data strobe signals.

For D3A I/Os, the IOM register bits connect to the I/O IDDQ pins. Furthermore, the I/O power-down register bits (PDD, PDR, and PD) fields have no relevance for D3A I/Os.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								AWDT
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	RVSEL	DQSNRST	DXSR[1:0]		DQSNRES[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
	DQSRES[3:0]				DXPDR	DXPDD	DXIOM	DXODT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – AWDT Active Window Data Train
Indicates if set that data training (DQS gate training and read valid training) should be performed with active DQS gate window. This is just for debug purposes. The default is to perform training with passive windowing.

Bit 15 – RVSEL ITMD Read Valid Select
Selects the scheme used for ITMD read valid.

Value	Description
0	ITMD read valid signal is generated by delayed DFI read enable signal.
1	ITMD read valid is generated by the ITMD itseld using asynchronous crossing.

Bit 14 – DQSNRST DQS# Reset
Indicates, if set, that the ITMS of DQS# should always be put in reset such that its output enable is always '1' and its data output is always '0'. This is done by driving the oe_set_b and do_rst_b pins of this ITMS to '0' in order to force the unused DQS# PAD to a known state of '0' in applications that don't use DQS# such as in LPDDR mode.

Bits 13:12 – DXSR[1:0] Data Slew Rate (D3F I/O Only)
Selects slew rate of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros.

Bits 11:8 – DQSNRES[3:0] DQS# Resistor
Selects the on-die pull-up/pull-down resistor for DQS# pins. Same encoding as DQSRES.
Note: DQS# resistor must be connected for LPDDR2.

Bits 7:4 – DQSRES[3:0] DQS Resistor
Selects the on-die pull-down/pull-up resistor for DQS pins.

Note: DQS resistor must be connected for LPDDR2.

DQSRES[3] selects pull-down (when set to 0) or pull-up (when set to 1).
DQSRES[2:0] selects the resistor value as follows:

Value	Description
000	Open: On-die resistor disconnected
001	688 ohms
010	611 ohms
011	550 ohms
100	500 ohms
101	458 ohms
110	393 ohms
111	344 ohms

Bit 3 – DXPDR Data Power Down Receiver
Powers down, when set, the input receiver on I/O for
DQ, DM, and DQS/DQS# pins of all DATX8 macros. It also powers down the PDQSR cell of all DAXT8 macros. This bit is ORed with the PDR and DQSRPD configuration bits of the individual DATX8.

Bit 2 – DXPDD Data Power Down Driver
Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the PDD configuration bit of the individual DATX8.

Bit 1 – DXIOM Data I/O Mode
Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the IOM configuration bit of the individual DATX8.

Bit 0 – DXODT Data On-Die Termination
Enables, when set, the on-die termination on the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the ODT configuration bit of the individual DATX8.

18.4.11 DDR3PHY DDR System General Configuration Register

Name: DDR3PHY_DSGCR
Offset: 0x2C
Reset: 0xFA00001F
Property: Read/Write

The DDR System General Configuration Register is used to configure miscellaneous features of the DDR system including the DFI 2.1 logic in the PUBL.

Bit	31	30	29	28	27	26	25	24
	CKEOE	RSTOE	ODTOE	CKOE	TPDOE	TPDPD	NL2OE	NL2PD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	0	1	0

Bit	23	22	21	20	19	18	17	16
	ODTPDD[3:0]				CKEPDD[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				FXDLAT	NOBUB	DQSGE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DQSGX[2:0]			LPDLLPD	LPIOPD	ZUEN	BDISEN	PUREN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

Bit 31 – CKEOE SDRAM CKE Output Enable
Enables, when set, the output driver on the I/O for SDRAM CKE pins.

Bit 30 – RSTOE SDRAM Reset Output Enable
Enables, when set, the output driver on the I/O for SDRAM RST# pin.

Bit 29 – ODTOE SDRAM ODT Output Enable
Enables, when set, the output driver on the I/O for SDRAM ODT pins.

Bit 28 – CKOE SDRAM CK Output Enable
Enables, when set, the output driver on the I/O for SDRAM CK/CK# pins.

Bit 27 – TPDOE SDRAM TPD Output Enable (LPDDR Only)
Enables, when set, the output driver on the I/O for SDRAM TPD pin.

Bit 26 – TPDPD SDRAM TPD Power Down Driver (LPDDR Only)
Powers down, when set, the output driver on the I/O for SDRAM TPD pin. Note that the power down of the receiver on the I/O for SDRAM TPD pin is controlled by ACIOCR[ACPDR] register bit.

Bit 25 – NL2OE Non-LPDDR2/LPDDR3 Output Enable
Enables, when set, the output driver on the I/O for non-LPDDR2/LPDDR3 (ODT, RAS#, CAS#, WE#, and BA) pins. This may be used when a chip that is designed for both LPDDR2/LPDDR3 and other DDR modes is being used in LPDDR2/LPDDR3 mode. For these pins, the I/O output enable signal (OE) is an AND of this bit and the respective output enable bit in ACIOCR or DSGCR registers.

Bit 24 – NL2PD Non-LPDDR2/LPDDR3 Power Down
Powers down, when set, the output driver and the input receiver on the I/O for non-LPDDR2/LPDDR3 (ODT, RAS#, CAS#, WE#, and BA) pins. This may be used when a chip that is designed for both LPDDR2/LPDDR3 and other DDR modes is being used in LPDDR2/LPDDR3 mode. For these pins, the I/O power down signal (PDD or PDR) is an OR of this bit and the respective power-down bit in ACIOCR register.

Bits 23:20 – ODTPDD[3:0] ODT Power Down Driver
Powers down, when set, the output driver on the I/O for ODT[3:0] pins. ODTPDD[0] controls the power down for ODT[0], ODTPDD[1] controls the power down for ODT[1], and so on.

Bits 19:16 – CKEPDD[3:0] CKE Power Down Driver
Powers down, when set, the output driver on the I/O for CKE[3:0] pins. CKEPDD[0] controls the power down for CKE[0], CKEPDD[1] controls the power down for CKE[1], and so on.

Bit 12 – FXDLAT Fixed Latency
Specifies whether all reads should be returned to the controller with a fixed read latency. Enabling fixed read latency increases the read latency.
If the design is compiled for HDR mode, then either NOBUB or FXDLAT must be set to 1.

Value	Description
0	Disable fixed read latency
1	Enable fixed read latency

Bit 11 – NOBUB No Bubbles
Specifies whether reads should be returned to the controller with no bubbles. Enabling no-bubble reads increases the read latency.
If the design is compiled for HDR mode, then either NOBUB or FXDLAT must be set to 1.

Value	Description
0	Bubbles are allowed during reads
1	Bubbles are not allowed during reads

Bits 10:8 – DQSGE[2:0] DQS Gate Early
Specifies the number of clock cycles for which the DQS gating must be enabled earlier than its normal position. Only applicable when using PDQSR I/O cell, passive DQS gating and no drift compensation. This field is recommended to be set to zero for all DDR types other than LPDDR2/LPDDR3. For LPDDR2/LPDDR3 it should be set to (tDQSCKmax - tDQSCKmin) divided by clock period and rounded up. tDQSCKmax and tDQSCKmin can be found in the LPDDR2 vendor datasheet.

Bits 7:5 – DQSGX[2:0] DQS Gate Extension
Specifies the number of clock cycles for which the DQS gating must be extended beyond the normal burst length width. Only applicable when using PDQSR I/O cell, passive DQS gating and no drift compensation. This field is recommended to be set to zero for all DDR types other than LPDDR2/LPDDR3. For LPDDR2/LPDDR3 it should be set to (tDQSCKmax. - tDQSCKmin), divided by clock period and rounded up. tDQSCKmax and tDQSCKmin can be found in the LPDDR2 vendor datasheet.

Bit 4 – LPDLLPD Low Power DLL Power Down
Specifies if set that the PHY should respond to the DFI low power opportunity request and power down the DLL of the PHY if the wakeup time request satisfies the DLL lock time.

Bit 3 – LPIOPD Low Power I/O Power Down
Specifies if set that the PHY should respond to the DFI low power opportunity request and power down the I/Os of the PHY.

Bit 2 – ZUEN Impedance Update Enable
Specifies if set that the PHY should perform impedance calibration (update) whenever there is a controller initiated DFI update request. Otherwise the PHY will ignore an update request from the controller.

Bit 1 – BDISEN Byte Disable Enable
Specifies if set that the PHY should respond to DFI byte disable request. Otherwise the byte disable from the DFI is ignored in which case bytes can only be disabled using the DXnGCR register.

Bit 0 – PUREN PHY Update Request Enable
Specifies if set, that the PHY should issue PHYinitiated DFI update request when there is DQS drift of more than ¾ of a clock cycle within one continuous (back-to-back) read burst. By default the PHY issues PHYinitiated update requests and the controller should respond otherwise the PHY may return erroneous values. The option to disable it is provided only for silicon evaluation and testing.

18.4.12 DDR3PHY DRAM Configuration Register

Name: DDR3PHY_DCR
Offset: 0x30
Reset: 0x0000000B
Property: Read/Write

This register is used to configure the DRAM system.

Bit	31	30	29	28	27	26	25	24
	TPD	RDIMM	UDIMM	DDR2T	NOSRA			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DDRTYPE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	MPRDQ	PDQ[2:0]			DDR8BNK	DDRMD[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	1

Bit 31 – TPD Test Power Down (LPDDR Only)
If set will place the DRAM in Deep Power Down mode.

Bit 30 – RDIMM Registered DIMM
Indicates if set that a registered DIMM is used. In this case, the PUBL increases the SDRAM write and read latencies (WL/RL) by 1. This only applies to PUBL internal SDRAM transactions. Transactions generated by the controller must make its own adjustments to WL/RL when using a registered DIMM.

Bit 29 – UDIMM Un-buffered DIMM Address Mirroring
Indicates if set that there is address mirroring on the second rank of an un-buffered DIMM (the rank connected to CS#[1]). In this case, the PUBL re-scrambles the bank and address when sending mode register commands to the second rank. This only applies to PUBL internal SDRAM transactions. Transactions generated by the controller must make its own adjustments when using an un-buffered DIMM. DCR[NOSRA] must be set if address mirroring is enabled.

Bit 28 – DDR2T DDR 2T Timing
Indicates if set that 2T timing should be used by PUBL internally generated SDRAM transactions.

Bit 27 – NOSRA No Simultaneous Rank Access
Specifies if set that simultaneous rank access on the same clock cycle is not allowed. This means that multiple chip select signals should not be asserted at the same time. This may be required on some DIMM systems.

Bits 9:8 – DDRTYPE[1:0] DDR Type
Selects the DDR type for the specified LPDDR mode.
Valid values for LPDDR2 are:

Value	Description
00	LPDDR2-S4
01	LPDDR2-S2

Value	Description
10	Reserved
11	Reserved

Bit 7 – MPRDQ Multi-Purpose Register (MPR) DQ (DDR3 Only)
Specifies the value that is driven on non-primary DQ pins during MPR reads.

Value	Description
0	Primary DQ drives out the data from MPR (0-1-0-1), non-primary DQs drive ‘0’.
1	Primary DQ and non-primary DQs all drive the same data from MPR (0-1-0-1).

Bits 6:4 – PDQ[2:0] Primary DQ (DDR3 Only)
Specifies the DQ pin in a byte that is designated as a primary pin for Multi-Purpose Register (MPR) reads. Valid values are 0 to 7 for DQ[0] to DQ[7], respectively.

Bit 3 – DDR8BNK DDR 8-Bank
Indicates if set that the SDRAM used has 8 banks. tRPA = tRP+1 and tFAW are used for 8-bank DRAMs, other tRPA = tRP and no tFAW is used. Note that a setting of 1 for DRAMs that have fewer than 8 banks still results in correct functionality but less tighter DRAM command spacing for the parameters described here.

Bits 2:0 – DDRMD[2:0] DDR Mode
SDRAM DDR mode.

Value	Description
001	DDR
010	DDR 2
011	DDR 3
100	LPDDR2 (Mobile DDR2)
101	LPDDR3 (Mobile DDR3)
110 - 111	Reserved

18.4.13 DDR3PHY DRAM Timing Parameters Register 0

Name: DDR3PHY_DTPR0
Offset: 0x34
Reset: 0x3092666E
Property: Read/Write

Various timings for the SDRAM are programmable for compatibility with different speed grades or different vendors. The tables describe the parameters that can be programmed through the SDRAM Timing Parameters Registers. These timing parameters are in clock cycles and are derived from corresponding parameters in the SDRAM datasheet divided by the clock cycle time. Non-integer values should be rounded up to the next integer. All the default values s are in decimal format and are for a 400MHz 1Gb x 16 DDR2800E SDRAM. Refer to the SDRAM datasheet for detailed description about these timing parameters.

Bit	31	30	29	28	27	26	25	24
	tCCD	tRC[5:0]						tRRD[3]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	tRRD[2:0]			tRAS[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	tRCD[3:0]				tRP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	1	1	0

Bit	7	6	5	4	3	2	1	0
	tWTR[2:0]			tRTP[2:0]			tMRD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	1	1	1	0

Bit 31 – tCCD Read to read and write to write command delay

Value	Description
0	BL/2 for DDR2 and 4 for DDR3
1	BL/2 + 1 for DDR2 and 5 for DDR3

Bits 30:25 – tRC[5:0] Activate to activate command delay (same bank)
Activate to activate command delay (same bank). Valid values are 2 to 42.

Bits 24:21 – tRRD[3:0] Activate to activate command delay (different banks)
Activate to activate command delay (different banks). Valid values are 1 to 8.

Bits 20:16 – tRAS[4:0] Activate to precharge command delay.
Valid values are 2 to 31.

Bits 15:12 – tRCD[3:0] Activate to read or write delay.
Minimum time from when an activate command is issued to when a read or write to the activated row can be issued. Valid values are 2 to 11.

Bits 11:8 – tRP[3:0] Precharge command period
The minimum time between a precharge command and any other command. Note that the Controller automatically derives tRPA for 8bank DDR2 devices by adding 1 to tRP. Valid values are 2 to 11.

Bits 7:5 – tWTR[2:0] Internal write to read command delay.
Internal write to read command delay. Valid values are 1 to 6.

Bits 4:2 – tRTP[2:0] Internal read to precharge command delay.
Internal read to precharge command delay. Valid values are 2 to 6. Note that even though RTP does not apply to JEDEC DDR devices, this parameter must still be set to a minimum value of 2 for DDR because the Controller always uses the DDR2 equation, AL + BL/2 + max(RTP,2) – 2, to compute the read to precharge timing (which is BL/2 for JEDEC DDR).

Bits 1:0 – tMRD[1:0] Load mode cycle time
The minimum time between a load mode register command and any other command. For DDR3 this is the minimum time between two load mode register commands. Valid values for DDR2 are 2 to 3. For DDR3, the value used for tMRD is 4 plus the value programmed in these bits, i.e. tMRD value for DDR3 ranges from 4 to 7. For LPDDR3, the value used for tMRD is 8 plus the value programmed in these bits.

18.4.14 DDR3PHY DRAM Timing Parameter Register 1

Name: DDR3PHY_DTPR1
Offset: 0x38
Reset: 0x09830090
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			tDQSCKmax[2:0]			tDQSCKmin[2:0]		
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	1	0	0	1

Bit	23	22	21	20	19	18	17	16
	tRFC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
					tRTODT	tMOD[1:0]		tFAW[5]
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	tFAW[4:0]					tRTW	tAOND/ tAOFD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	0	0	0	0

Bits 29:27 – tDQSCKmax[2:0] Maximum DQS output access time from CK/CK# (LPDDR2 only)
This value is used for implementing read-to-write spacing. Valid values are 1 to 7.

Bits 26:24 – tDQSCKmin[2:0] DQS output access time from CK/CK# (LPDDR2/3 only)
This value is used for computing the read latency. Valid values are 0 to 7. This value is derived from the corresponding parameter in the SDRAM datasheet divided by the clock cycle time without rounding up. The fractional remainder is automatically adjusted for by data training in quarter clock cycle units. If data training is not performed then this fractional remainder must be converted to quarter clock cycle units and the gating registers (DXnDQSTR) adjusted accordingly.

Bits 23:16 – tRFC[7:0] Refresh-to-Refresh
Indicates the minimum time, in clock cycles, between two refresh commands or between a refresh and an active command. This is derived from the minimum refresh interval from the datasheet, tRFC(min), divided by the clock cycle time. The default number of clock cycles is for the largest JEDEC tRFC(min parameter value supported.

Bit 11 – tRTODT Read to ODT delay (DDR3 only)
Specifies whether ODT can be enabled immediately after the read post-amble or one clock delay has to be added. If tRTODT is set to 1, then the read-to-write latency is increased by 1 if ODT is enabled.

Value	Description
0	ODT may be turned on immediately after read post-amble
1	ODT may not be turned on until one clock after the read post-amble

Bits 10:9 – tMOD[1:0] Load mode update delay (DDR3 only)
The minimum time between a load mode register command and a non-load mode register command.

Value	Description
00	12
01	13
10	14
11	15

Bits 8:3 – tFAW[5:0] 4-bank activate period
No more than 4-bank activate commands may be issued in a given tFAW period. Only applies to 8-bank devices. Valid values are 2 to 31.

Bit 2 – tRTW Read to Write command delay
This parameter allows the user to increase the delay between issuing Write commands to the SDRAM when preceded by Read commands. This provides an option to increase bus turn-around margin for high frequency systems.

Value	Description
0	Standard bus turn around delay
1	Adds 1 clock to standard bus turn around delay

Bits 1:0 – tAOND/ tAOFD[1:0] ODT turn-on/turn-off delays (DDR2 only)
The delays are in clock cycles.
Most DDR2 devices utilize a fixed value of 2/2.5. For non-standard SDRAMs, the user must ensure that the operational Write Latency is always greater than or equal to the ODT turn-on delay. For example, a DDR2 SDRAM with CAS latency set to 3 and CAS additive latency set to 0 has a Write Latency of 2. Thus 2/2.5 can be used, but not 3/3.5 or higher.

Value	Description
00	2/2.5
01	3/3.5
10	4/4.5
11	5/5.5

18.4.15 DDR3PHY DRAM Timing Parameter Register 2

Name: DDR3PHY_DTPR2
Offset: 0x3C
Reset: 0x1001A0C8
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				tDLLK[9:5]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	tDLLK[4:0]					tCKE[3:1]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	tCKE[0]	tXP[4:0]					tXS[9:8]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	tXS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	1	0	0	0

Bits 28:19 – tDLLK[9:0] DLL locking time
Valid values are 2 to 1023.

Bits 18:15 – tCKE[3:0] CKE minimum pulse width
Also specifies the minimum time that the SDRAM must remain in Power-down or Self-refresh mode. For DDR3 this parameter must be set to the value of tCKESR which is usually bigger than the value of tCKE. Valid values are 2 to 15.

Bits 14:10 – tXP[4:0] Power down exit delay
The minimum time between a power down exit command and any other command. This parameter must be set to the maximum of the various minimum power down exit delay parameters specified in the SDRAM datasheet, i.e. max(tXP, tXARD, tXARDS) for DDR2 and max(tXP, tXPDLL) for DDR3. Valid values are 2 to 31.

Bits 9:0 – tXS[9:0] Self refresh exit delay
The minimum time between a self refresh exit command and any other command. This parameter must be set to the maximum of the various minimum self refresh exit delay parameters specified in the SDRAM datasheet, i.e. max(tXSNR, tXSRD) for DDR2 and max(tXS, tXSDLL) for DDR3. Valid values are 2 to 1023.

18.4.16 DDR3PHY Mode Register 0 (MR0) (DDR3 Mode)

Name: DDR3PHY_MR0_DDR3
Offset: 0x40
Reset: 0x00000294
Property: Read/Write

SDRAM definitions controlled by the Mode Register 0 (MR0) include, among other things, burst length, burst type, CAS latency, operating (normal or test) mode, DLL reset, write recovery, and power-down modes. The PUBL Mode Register 0 (MR0) maps to DDR3 Mode Register 0 (MR0), DDR2 Mode Register (MR) and DDR Mode Register (MR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				PD		WR[2:0]		DR
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	TM	CL3	CL2	CL1	BT	CL0	BL[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	0	1

Bit 12 – PD Power-Down Control
Controls the exit time for power-down modes. Refer to the SDRAM datasheet for details on power-down modes.

Value	Description
0	Slow exit (DLL off)
1	Fast exit (DLL on)

Bits 11:9 – WR[2:0] Write Recovery
This is the value of the write recovery in clock cycles. It is calculated by dividing the datasheet write recovery time, tWR (ns) by the datasheet clock cycle time, tCK (ns) and rounding up a non-integer value to the next integer.
Note: tWR (ns) is the time from the first SDRAM positive clock edge after the last data-in pair of a write command, to when a precharge of the same bank can be issued.
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
001	5
010	6
011	7
100	8
101	10
110	12

Bit 8 – DR DLL Reset
Writing a ‘1’ to this bit will reset the SDRAM DLL. This bit is selfclearing, i.e. it returns back to ‘0’ after the DLL reset has been issued.

Bit 7 – TM Operating Mode
Selects either normal operating mode (0) or test mode (1). Test mode is reserved for the manufacturer and should not be used.

Bit 6 – CL3 CAS Latency
The delay, in clock cycles, between when the SDRAM registers a read command to when data is available.
CL3, CL2, CL1, CL0= MR0 [6:4,2]
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
0010	5
0100	6
0110	7
1000	8
1010	9
1100	10
1110	11

Bit 5 – CL2 CAS Latency
See CL3.
CL = MR0 [6:4,2]

Bit 4 – CL1 CAS Latency
See CL3.
CL = MR0 [6:4,2]

Bit 3 – BT Burst Type
Indicates whether a burst is sequential or interleaved.

Value	Description
0	Sequential burst
1	Interleaved burst

Bit 2 – CL0 CAS Latency
See CL3.
CL = MR0 [6:4,2]

Bits 1:0 – BL[1:0] Burst Length
Determines the maximum number of column locations that can be accessed during a given read or write command.
Valid values are as follows.

Value	Description
00	8 (fixed)
01	4 or 8 (on the fly)
10	4 (fixed)
11	Reserved

18.4.17 DDR3PHY Mode Register 0 (MR0) (DDR2 Mode)

Name: DDR3PHY_MR0_DDR2
Offset: 0x40
Reset: 0x00000A52
Property: Read/Write

SDRAM definitions controlled by the Mode Register 0 (MR0) include, among other things, burst length, burst type, CAS latency, operating (normal or test) mode, DLL reset, write recovery, and power-down modes. The PUBL Mode Register 0 (MR0) maps to DDR3 Mode Register 0 (MR0), DDR2 Mode Register (MR) and DDR Mode Register (MR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				PD	WR[2:0]			DR
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	TM	CL[2:0]			BT	BL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	1	0

Bit 12 – PD Power-Down Control
Controls the exit time for power-down modes. Refer to the SDRAM datasheet for details on power-down modes.

Value	Description
0	Fast exit
1	Slow exit

Bits 11:9 – WR[2:0] Write Recovery
This is the value of the write recovery in clock cycles. It is calculated by dividing the datasheet write recovery time, tWR (ns) by the datasheet clock cycle time, tCK (ns) and rounding up a non-integer value to the next integer.
Note: tWR (ns) is the time from the first SDRAM positive clock edge after the last data-in pair of a write command, to when a precharge of the same bank can be issued.
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
001	2
010	3
011	4
100	5
101	6

Bit 8 – DR DLL Reset
Writing a ‘1’ to this bit will reset the SDRAM DLL. This bit is selfclearing, i.e. it returns back to ‘0’ after the DLL reset has been issued.

Bit 7 – TM Operating Mode
Selects either normal operating mode (0) or test mode (1). Test mode is reserved for the manufacturer and should not be used.

Bits 6:4 – CL[2:0] CAS Latency
The delay, in clock cycles, between when the SDRAM registers a read command to when data is available. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
010	2
011	3
100	4
101	5
110	6

Bit 3 – BT Burst Type
Indicates whether a burst is sequential or interleaved.

Value	Description
0	Sequential burst
1	Interleaved burst

Bits 2:0 – BL[2:0] Burst Length
Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
010	4
011	8

18.4.18 DDR3PHY Mode Register 0 (MR0) (DDR Mode)

Name: DDR3PHY_MR0_DDR
Offset: 0x40
Reset: 0x00000A52
Property: Read/Write

SDRAM definitions controlled by the Mode Register 0 (MR0) include, among other things, Burst Length, Burst Type, CAS Latency, Operating (Normal or Test) mode, DLL Reset, Write Recovery, and Power-down modes. The PUBL Mode Register 0 (MR0) maps to DDR3 Mode Register 0 (MR0), DDR2 Mode Register (MR) and DDR Mode Register (MR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DR
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	TM	CL[2:0]			BT	BL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	1	0

Bit 8 – DR DLL Reset
Writing a ‘1’ to this bit will reset the SDRAM DLL. This bit is selfclearing, i.e. it returns back to ‘0’ after the DLL reset has been issued.

Bit 7 – TM Operating Mode
Selects either normal operating mode (0) or test mode (1). Test mode is reserved for the manufacturer and should not be used.

Bits 6:4 – CL[2:0] CAS Latency
The delay, in clock cycles, between when the SDRAM registers a read command to when data is available. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
010	2
011	3
101	1.5 (Not supported by the PHY)
110	2.5 (Not supported by the PHY)

Bit 3 – BT Burst Type
Indicates whether a burst is sequential or interleaved.

Value	Description
0	Sequential burst
1	Interleaved burst

Bits 2:0 – BL[2:0] Burst Length
Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
001	2
010	4
011	8

18.4.19 DDR3PHY Mode Register 1 (MR1) (DDR3 Mode)

Name: DDR3PHY_MR1_DDR3
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

SDRAM definitions controlled by the Mode register 1 (MR1) include, among other things, DLL enable/disable, drive strength, on-die termination (ODT) resistance, CAS additive latency, off-chip driver (OCD) impedance calibration, DQS_b and RDQS/RDQS_b enable/disable, and output enable. The PUBL Mode Register 1 (MR1) maps to DDR3 Mode Register 1 (MR1), DDR2 Extended Mode Register (EMR), DDR Extended Mode Register (EMR), LPDDR2 Mode Register 1 (MR1), and LPDDR3 Mode Register 1 (MR1).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				QOFF	TDQS		RTT2	
Access				R/W	R/W		R/W	
Reset				0	0		0	

Bit	7	6	5	4	3	2	1	0
	LEVEL	RTT1	DIC1	AL[1:0]		RTT0	DIC0	DE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – QOFF Output Enable/Disable
When '0', all outputs function normally; when '1', all SDRAM outputs are disabled removing output buffer current. This feature is intended to be used for IDD characterization of read current and should not be used in normal operation.

Bit 11 – TDQS Termination Data Strobe
When enabled ('1') TDQS provides additional termination resistance outputs that may be useful in some system configurations. Refer to the SDRAM datasheet for details.

Bit 9 – RTT2 On Die Termination
Selects the effective resistance for SDRAM on die termination. RTT2-RTT1-RTT0: Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
000	ODT disabled
001	RZQ/4
010	RZQ/2
011	RZQ/6
100	RZQ/12
101	RZQ/8

Bit 7 – LEVEL Write Leveling Enable
Enables write-leveling when set.

Bit 6 – RTT1 On Die Termination
Selects the effective resistance for SDRAM on die termination. See RTT2.

Bit 5 – DIC1 Output Driver Impedance Control
Controls the output drive strength. DIC1-DIC0: Valid values are as follows.

Value	Description
00	RZQ/6
01	RZQ/7
10	Reserved
11	Reserved

Bits 4:3 – AL[1:0] Posted CAS Additive Latency
Setting additive latency that allows read and write commands to be issued to the SDRAM earlier than normal (refer to the SDRAM datasheet for details).

Value	Description
00	0 (AL disabled)
01	CL - 1
10	CL - 2
11	Reserved

Bit 2 – RTT0 On Die Termination
Selects the effective resistance for SDRAM on die termination. See RTT2.

Bit 1 – DIC0 Output Driver Impedance Control
See DIC1.

Bit 0 – DE DLL Enable/Disable
DLL must be enabled for normal operation.

Value	Description
0	Enable the DLL
1	Disable the DLL

18.4.20 DDR3PHY Mode Register 1 (MR1) (DDR2 Mode)

Name: DDR3PHY_MR1_DDR2
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

SDRAM definitions controlled by the Mode register 1 (MR1) include, among other things, DLL enable/disable, drive strength, on-die termination (ODT) resistance, CAS additive latency, off-chip driver (OCD) impedance calibration, DQS_b and RDQS/RDQS_b enable/disable, and output enable. The PUBL Mode Register 1 (MR1) maps to DDR3 Mode Register 1 (MR1), DDR2 Extended Mode Register (EMR), DDR Extended Mode Register (EMR), LPDDR2 Mode Register 1 (MR1), and LPDDR3 Mode Register 1 (MR1).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				QOFF	RDQS	DQS	OCD[2:1]	
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OCD[0]	RTT1	AL[2:0]			RTT0	DIC	DE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – QOFF Output Enable/Disable
When ‘0’, all outputs function normal, when ‘1’ all SDRAM outputs are disabled removing output buffer current. This feature is intended to be used for IDD characterization of read current and should not be used in normal operation.

Bit 11 – RDQS RDQS Enable/Disable
When enabled (‘1’), RDQS is identical in function and timing to data strobe DQS during a read, and ignored during a write. A ‘0’ disables the SDRAM from driving RDQS. The Controller does not allow the user to change this bit.

Bit 10 – DQS DQS_b Enable/Disable
When ‘0’, DQS_b is the complement of the differential data strobe pair DQS/DQS_b. When ‘1’, DQS is used in a single-ended mode and the DQS_b pin is disabled. Also used to similarly enable/disable RDQS_b if RDQS is enabled. The Controller does not allow the user to change this bit.

Bits 9:7 – OCD[2:0] Off-Chip Driver (OCD) Impedance Calibration
Used to calibrate and match pull-up to pull-down impedance to 18 Ω nominal (refer to the SDRAM datasheet for details). Note that OCD is not supported by all vendors. Refer to the SDRAM datasheet for details on the recommended OCD settings. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
`000`	OCD calibration mode exit
`001`	Drive (1) pull-up
`010`	Drive (0) pull-down
`100`	OCD enter adjust mode
`111`	OCD calibration default

Bit 6 – RTT1 On Die Termination
Selects the effective resistance for SDRAM on die termination.
RTT1-RTT0: Valid values are as follows.

Value	Description
00	ODT disabled
01	75 Ω
10	150 Ω
11	50 Ω (some vendors)

Bits 5:3 – AL[2:0] Posted CAS Additive Latency
Setting additive latency that allows read and write commands to be issued to the SDRAM earlier than normal (refer to the SDRAM datasheet for details). Valid values are as follows. All other settings are reserved and should not be used. The maximum allowed value of AL is tRCD-1.

Value	Description
000	0
001	1
010	2
011	3
100	4
101	5

Bit 2 – RTT0 On Die Termination
Selects the effective resistance for SDRAM on die termination. See RTT1.

Bit 1 – DIC Output Driver Impedance Control
Controls the output drive strength.

Value	Description
0	Full strength
1	Reduced strength

Bit 0 – DE DLL Enable/Disable
DLL must be enabled for normal operation.

Value	Description
0	Enable the DLL
1	Disable the DLL

18.4.21 DDR3PHY Mode Register 1 (MR1) (DDR Mode)

Name: DDR3PHY_MR1_DDR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

SDRAM definitions controlled by the Mode register 1 (MR1) include, among other things, DLL enable/disable, drive strength, on-die termination (ODT) resistance, CAS additive latency, off-chip driver (OCD) impedance calibration, DQS_b and RDQS/RDQS_b enable/disable, and output enable. The PUBL Mode Register 1 (MR1) maps to DDR3 Mode Register 1 (MR1), DDR2 Extended Mode Register (EMR), DDR Extended Mode Register (EMR), LPDDR2 Mode Register 1 (MR1), and LPDDR3 Mode Register 1 (MR1).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DS	DE
Access							R/W	R/W
Reset							0	0

Bit 1 – DS Output Drive Strength
Controls the output drive strength.

Value	Description
0	Normal
1	Weak

Bit 0 – DE DLL Enable/Disable
DLL must be enabled for normal operation.

Value	Description
0	Enable the DLL
1	Disable the DLL

18.4.22 DDR3PHY Mode Register 1 (MR1) (LPDDR2 Mode)

Name: DDR3PHY_MR1_LPDDR2
Offset: 0x44
Reset: 0x00000002
Property: Read/Write

SDRAM definitions controlled by the Mode register 1 (MR1) include, among other things, DLL enable/disable, drive strength, on-die termination (ODT) resistance, CAS additive latency, off-chip driver (OCD) impedance calibration, DQS_b and RDQS/RDQS_b enable/disable, and output enable. The PUBL Mode Register 1 (MR1) maps to DDR3 Mode Register 1 (MR1), DDR2 Extended Mode Register (EMR), DDR Extended Mode Register (EMR), LPDDR2 Mode Register 1 (MR1), and LPDDR3 Mode Register 1 (MR1).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	nWR[2:0]			WC	BT	BL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bits 7:5 – nWR[2:0] Write Recovery
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
001	3
010	4
011	5
100	6
101	7
110	8

Bit 4 – WC Wrap Control

Value	Description
0	Wrap
1	No wrap

Bit 3 – BT Burst Type
Indicates whether a burst is sequential or interleaved.

Value	Description
0	Sequential burst
1	Interleaved burst

Bits 2:0 – BL[2:0] Burst Length
Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
010	4
011	8
100	16

18.4.23 DDR3PHY Mode Register 1 (MR1) (LPDDR3 Mode)

Name: DDR3PHY_MR1_LPDDR3
Offset: 0x44
Reset: 0x00000002
Property: Read/Write

SDRAM definitions controlled by the Mode register 1 (MR1) include, among other things, DLL enable/disable, drive strength, on-die termination (ODT) resistance, CAS additive latency, off-chip driver (OCD) impedance calibration, DQS_b and RDQS/RDQS_b enable/disable, and output enable. The PUBL Mode Register 1 (MR1) maps to DDR3 Mode Register 1 (MR1), DDR2 Extended Mode Register (EMR), DDR Extended Mode Register (EMR), LPDDR2 Mode Register 1 (MR1), and LPDDR3 Mode Register 1 (MR1).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		nWR[2:0]					BL[2:0]	
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	1	0

Bits 7:5 – nWR[2:0] Write Recovery
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
	If nWRE (MR2 [4]) = 0:
001B	nWR=3
100B	nWR=6
110B	nWR=8
111B	nWR=9
	If nWRE (MR2[4]) = 1:
000B	nWR=10
001B	nWR=11
010B	nWR=12

Bits 2:0 – BL[2:0] Burst Length
Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are: 011 = 8 All other settings are reserved and should not be used.

18.4.24 DDR3PHY Mode Register 2 (MR2/EMR2) (DDR3 Mode)

Name: DDR3PHY_MR2_DDR3
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Mode Register 2 controls among other things refresh related features of SDRAMs. The PUBL Mode Register 2 (MR2) maps to DDR3 Mode Register 2 (MR2), DDR2 Extended Mode Register 2 (EMR2), LPDDR2 Mode Register 2 (MR2) and LPDDR3 Mode Register 2 (MR2).

Note that LPDDR EMR register is mapped to the PUBL MR2 register because the SDRAM address for the this register is BA[1:0] = 2'b10.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSVD[4:0]					RTTWR[1:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
	SRT	ASR	CWL[2:0]			PASR[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:11 – RSVD[4:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’.

Bits 10:9 – RTTWR[1:0] Dynamic ODT
Selects RTT for dynamic ODT.

Value	Description
`00`	Dynamic ODT off
`01`	RZQ/4
`10`	RZQ/2
`11`	Reserved

Bit 7 – SRT Self-Refresh Temperature Range
Selects either normal or extended operating temperature range during self-refresh.

Value	Description
`0`	Normal self-refresh temperature range
`1`	Extended self-refresh temperature range

Bit 6 – ASR Auto Self-Refresh
When enabled (‘1’), SDRAM automatically provides self-refresh power management functions for all supported operating temperature values.
Otherwise the SRT bit must be programmed to indicate the temperature range.

Bits 5:3 – CWL[2:0] CAS Write Latency
The delay, in clock cycles, between when the SDRAM registers a write command to when write data is available. Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
000	5 (tCK = 2.5 ns)
001	6 (2.5 ns > tCK = 1.875 ns)
010	7 (1.875 ns > tCK = 1.5 ns)
011	8 (1.5 ns > tCK = 1.25 ns)

Bits 2:0 – PASR[2:0] Partial Array Self Refresh
Specifies that data located in areas of the array beyond the specified location will be lost if self refresh is entered.

Value	Description
	Valid settings for 4 banks:
000	Full Array
001	Half Array (BA[1:0] = 00 & 01)
010	Quarter Array (BA[1:0] = 00)
011	Not defined
100	3/4 Array (BA[1:0] = 01, 10, & 11)
101	Half Array (BA[1:0] = 10 & 11)
110	Quarter Array (BA[1:0] = 11)
111	Not defined
	Valid settings for 8 banks:
000	Full Array
001	Half Array (BA[2:0] = 000, 001, 010 & 011)
010	Quarter Array (BA[2:0] = 000, 001)
011	1/8 Array (BA[2:0] = 000)
100	3/4 Array (BA[2:0] = 010, 011, 100, 101, 110 & 111)
101	Half Array (BA[2:0] = 100, 101, 110 & 111)
110	Quarter Array (BA[2:0] = 110 & 111)
111	1/8 Array (BA[2:0] 111)

18.4.25 DDR3PHY Mode Register 2 (MR2/EMR2) (DDR2 Mode)

Name: DDR3PHY_MR2_DDR2
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Mode Register 2 controls among other things refresh related features of SDRAMs. The PUBL Mode Register 2 (MR2) maps to DDR3 Mode Register 2 (MR2), DDR2 Extended Mode Register 2 (EMR2), LPDDR2 Mode Register 2 (MR2) and LPDDR3 Mode Register 2 (MR2).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSVD[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	7	6	5	4	3	2	1	0
	SRF				DCC	PASR[2:0]		
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bits 15:11 – RSVD[4:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’

Bit 7 – SRF Self Refresh Rate
Enables, if set, high temperature self refresh rate.

Bit 3 – DCC Duty Cycle Corrector
Enables, if set, duty cycle correction within SDRAM.

Bits 2:0 – PASR[2:0] Partial Array Self Refresh
Specifies that data located in areas of the array beyond the specified location will be lost if self refresh is entered.

Value	Description
	Valid settings for 4 banks:
000	Full Array
001	Half Array (BA[1:0] = 00 & 01)
010	Quarter Array (BA[1:0] = 00)
011	Not defined
100	3/4 Array (BA[1:0] = 01, 10, & 11)
101	Half Array (BA[1:0] = 10 & 11)
110	Quarter Array (BA[1:0] = 11)
111	Not defined
	Valid settings for 8 banks:
000	Full Array
001	Half Array (BA[2:0] = 000, 001, 010 & 011)
010	Quarter Array (BA[2:0] = 000, 001)
011	1/8 Array (BA[2:0] = 000)

Value	Description
100	3/4 Array (BA[2:0] = 010, 011, 100, 101, 110 & 111)
101	Half Array (BA[2:0] = 100, 101, 110 & 111)
110	Quarter Array (BA[2:0] = 110 & 111)
111	1/8 Array (BA[2:0] 111)

18.4.26 DDR3PHY Mode Register 2 (MR2/EMR2) (LPDDR2 Mode)

Name: DDR3PHY_MR2_LPDDR2
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Mode Register 2 controls among other things refresh related features of SDRAMs. The PUBL Mode Register 2 (MR2) maps to DDR3 Mode Register 2 (MR2), DDR2 Extended Mode Register 2 (EMR2), LPDDR2 Mode Register 2 (MR2) and LPDDR3 Mode Register 2 (MR2).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RSVD[3:0]				RL/WL[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – RSVD[3:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’

Bits 3:0 – RL/WL[3:0] Read and Write Latency
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
0001	RL = 3 /WL = 1
0010	RL = 4 / WL = 2
0011	RL = 5 / WL = 2
0100	RL = 6 / WL = 3
0101	RL = 7 / WL = 4
0110	RL = 8 / WL = 4

18.4.27 DDR3PHY Mode Register 2 (MR2/EMR2) (LPDDR3 Mode)

Name: DDR3PHY_MR2_LPDDR3
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Mode Register 2 controls among other things refresh related features of SDRAMs. The PUBL Mode Register 2 (MR2) maps to DDR3 Mode Register 2 (MR2), DDR2 Extended Mode Register 2 (EMR2), LPDDR2 Mode Register 2 (MR2) and LPDDR3 Mode Register 2 (MR2).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RSVD[3:0]				RL/WL[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – RSVD[3:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’.

Bits 3:0 – RL/WL[3:0] Read and Write Latency
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
`0001`	RL = 3 /WL = 1
`0010`	RL = 4 / WL = 2
`0011`	RL = 5 / WL = 2
`0100`	RL = 6 / WL = 3
`0101`	RL = 7 / WL = 4
`0110`	RL = 8 / WL = 4

18.4.28 DDR3PHY Mode Register 3 (MR3) (DDR3 Mode)

Name: DDR3PHY_MR3_DDR3
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Mode Register 3 controls, among other things, the multi-purpose register. The PUBL Mode Register 3 (MR3) maps to DDR3 Mode Register 3 (MR3), DDR2 Extended Mode Register 3 (EMR3), LPDDR2 Mode Register 3 (MR3), and LPDDR3 Mode Register 3 (MR3).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSVD[12:5]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RSVD[4:0]					MPR	MPRLOC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:3 – RSVD[12:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’

Bit 2 – MPR Multi-Purpose Register Enable
Enables, if set, that read data should come from the Multi-Purpose Register. Otherwise read data come from the DRAM array.

Bits 1:0 – MPRLOC[1:0] Multi-Purpose Register (MPR) Location
Selects MPR data location. Valid value is:
00 = Predefined pattern for system calibration
All other settings are reserved and should not be used.

18.4.29 DDR3PHY Mode Register 3 (MR3) (DDR2 Mode)

Name: DDR3PHY_MR3_EMR3_DDR2
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Mode Register 3 controls, among other things, the multi-purpose register. The PUBL Mode Register 3 (MR3) maps to DDR3 Mode Register 3 (MR3), DDR2 Extended Mode Register 3 (EMR3), LPDDR2 Mode Register 3 (MR3), and LPDDR3 Mode Register 3 (MR3).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSVD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RSVD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RSVD[15:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to '0'.

18.4.30 DDR3PHY Mode Register 3 (MR3) (LPDDR2 Mode)

Name: DDR3PHY_MR3_LPDDR2
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Mode Register 3 controls, among other things, the multi-purpose register. The PUBL Mode Register 3 (MR3) maps to DDR3 Mode Register 3 (MR3), DDR2 Extended Mode Register 3 (EMR3), LPDDR2 Mode Register 3 (MR3), and LPDDR3 Mode Register 3 (MR3).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RSVD[3:0]				DS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – RSVD[3:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’.

Bits 3:0 – DS[3:0] Drive Strength.
Valid values are as follows. All other settings are reserved and should not be used.

Value	Description
0000	Reserved
0001	34.3-ohm typical
0010	40-ohm typical
0011	48-ohm typical
0100	60-ohm typical
0101	Reserved
0110	80-ohm typical
0111	120-ohm typical

18.4.31 DDR3PHY Mode Register 3 (MR3) (LPDDR3 Mode)

Name: DDR3PHY_MR_LPDDR3
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Mode Register 3 controls, among other things, the multi-purpose register. The PUBL Mode Register 3 (MR3) maps to DDR3 Mode Register 3 (MR3), DDR2 Extended Mode Register 3 (EMR3), LPDDR2 Mode Register 3 (MR3), and LPDDR3 Mode Register 3 (MR3).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RSVD[3:0]				PDCTL[1:0]		DQODT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – RSVD[3:0] Reserved
These are JEDEC reserved bits and are recommended by JEDEC to be programmed to ‘0’.

Bits 3:2 – PDCTL[1:0] Power Down control

Value	Description
0	ODT disabled by DRAM during power
1	ODT enabled by DRAM during power down

Bits 1:0 – DQODT[1:0] On Die Termination

Value	Description
00	Disable (default)
01	Rzq/4 (optional for LPDDR3- 1066 devices)
10	Rzq/2
11	Rzq/1

18.4.32 DDR3PHY ODT Configuration Register

Name: DDR3PHY_ODTCR
Offset: 0x50
Reset: 0x84210000
Property: Read/Write

This register configures how ODT should be controlled on the different ranks when writing or reading a particular rank. This provides full flexibility on how the overall termination of the SDRAM system is set depending on how many ranks are used and whether the DIMM slots are occupied or not.

Bit	31	30	29	28	27	26	25	24
	WRODT3[3:0]				WRODT2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	1	0	0

Bit	23	22	21	20	19	18	17	16
	WRODT1[3:0]				WRODT0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	RDODT3[3:0]				RDODT2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDODT1[3:0]				RDODT0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – WRODT3[3:0] WRODT3

Bits 27:24 – WRODT2[3:0] WRODT2

Bits 23:20 – WRODT1[3:0] WRODT1

Bits 19:16 – WRODT0[3:0] Write ODT
Specifies whether ODT should be enabled (‘1’) or disabled (‘0’) on each of the up to four ranks when a write command is sent to rank n. WRODT0, WRODT1, WRODT2, and WRODT3 specify ODT settings when a write is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to enable ODT only on rank being written to.

Bits 15:12 – RDODT3[3:0] RDODT3

Bits 11:8 – RDODT2[3:0] RDODT2

Bits 7:4 – RDODT1[3:0] RDODT1

Bits 3:0 – RDODT0[3:0] Read ODT
Specifies whether ODT should be enabled (‘1’) or disabled (‘0’) on each of the up to four ranks when a read command is sent to rank n. RDODT0, RDODT1, RDODT2, and RDODT3 specify ODT settings when a read is to rank 0, rank 1, rank 2, and rank 3, respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 3. Default is to disable ODT during reads.
Note: The RODT bits are applicable to DDR2 only. Since the PUBL does not support DIMM configuration while in DDR3 mode, the Rank ODT signal is not asserted for idle ranks in a multi-rank configuration.

18.4.33 DDR3PHY Data Training Address Register

Name: DDR3PHY_DTAR
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

This register optionally sets the address used during data training.

The data training address is based on a decoded bank/row/column address.

The user should ensure the column address starts at the beginning of a burst-length-16 boundary (A[3:0] = DTCOL[3:0] = 0000).

Bit	31	30	29	28	27	26	25	24
	DTMPR	DTBANK[2:0]			DTROW[15:12]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DTROW[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DTROW[3:0]				DTCOL[11:8]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DTCOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – DTMPR Data Training Using MPR (DDR3 Only)
Specifies, if set, that data-training should use the SDRAM Multi-Purpose Register (MPR) register. Otherwise data-training is performed by first writing to some locations in the SDRAM and then reading them back.

Bits 30:28 – DTBANK[2:0] Data Training Bank Address
Selects the SDRAM bank address to be used during data training.

Bits 27:12 – DTROW[15:0] Data Training Row Address
Selects the SDRAM row address to be used during data training.

Bits 11:0 – DTCOL[11:0] Data Training Column Address
Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be “0000”.

18.4.34 DDR3PHY Data Training Data Register 0

Name: DDR3PHY_DTDR0
Offset: 0x58
Reset: 0xDD22EE11
Property: Read/Write

These registers optionally set the 8 data bytes used during data training. Data training uses a total burst length of 8, i.e. one access for burst length of 8, two accesses for burst length of 4 and four accesses for burst length of 2. The data set in DTDR registers is used as the data for each of the 8 data cycles. The data for the first 4 clock cycles (DTBYTE0 to DTBYTE3) come from DTDR0 register, and the data for the last 4 cycles (DTBYT4 to DTBYTE7) come from DTDR1 register.

Bit	31	30	29	28	27	26	25	24
	DTBYTE3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	1	1	1	0	1
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	DTBYTE2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	1	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	DTBYTE1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	1	1	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								DTBYTE0
Access								R/W
Reset								1

Bits 31:24 – DTBYTE3[7:0] Data Training Data
The first 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bits 23:16 – DTBYTE2[7:0] Data Training Data
The first 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bits 15:8 – DTBYTE1[7:0] Data Training Data
The first 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bit 0 – DTBYTE0 Data Training Data
The first 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

18.4.35 DDR3PHY Data Training Data Register 1

Name: DDR3PHY_DTDR1
Offset: 0x5C
Reset: 0x7788BB44
Property: Read/Write

These registers optionally set the 8 data bytes used during data training. Data training uses a total burst length of 8, i.e. one access for burst length of 8, two accesses for burst length of 4 and four accesses for burst length of 2. The data set in DTDR registers is used as the data for each of the 8 data cycles. The data for the first 4 clock cycles (DTBYTE0 to DTBYTE3) come from DTDR0 register, and the data for the last 4 cycles (DTBYT4 to DTBYTE7) come from DTDR1 register.

Bit	31	30	29	28	27	26	25	24
	DTBYTE7[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	1	1
Bit	23	22	21	20	19	18	17	16
	DTBYTE6[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	0	0	0
Bit	15	14	13	12	11	10	9	8
	DTBYTE5[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	1	1	0	1	1
Bit	7	6	5	4	3	2	1	0
	DTBYTE4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

Bits 31:24 – DTBYTE7[7:0] Data Training Data
The second 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bits 23:16 – DTBYTE6[7:0] Data Training Data
The second 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bits 15:8 – DTBYTE5[7:0] Data Training Data
The second 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

Bits 7:0 – DTBYTE4[7:0] Data Training Data
The second 4 bytes of data used during data training. This same data byte is used for each byte lane. Default sequence is a walking 1 while toggling data every data cycle.

18.4.36 DDR3PHY DCU Address Register

Name: DDR3PHY_DCUAR
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

DCU Address Register selects the DCU cache and the address of the cache that should be accessed when subsequent configuration writes or reads to the caches is triggered using the DCU Data Register(DCUDR). A write to the DCUAR will automatically transfer the first cache data word into the DCU Data Register (DCUDR) and optionally increment the address if the cache assess is a read.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ATYPE	INCA	CSEL[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CSADDR[3:0]				CWADDR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – ATYPE Access Type
Specifies the type of access to be performed using this address.

Value	Description
0	Write access
1	Read access

Bit 10 – INCA Increment Address
Specifies, if set, that the cache address specified in WADDR and SADDR should be automatically incremented after each access of the cache. The increment happens in such a way that all the slices of a selected word are first accessed before going to the next word.

Bits 9:8 – CSEL[1:0] Cache Select
Selects the cache to be accessed.

Value	Description
00	Command cache
01	Expected data cache
10	Read data cache
11	Reserved

Bits 7:4 – CSADDR[3:0] Cache Slice Address
Address of the cache slice to be accessed.

Bits 3:0 – CWADDR[3:0] Cache Word Address
Address of the cache word to be accessed.

18.4.37 DDR3PHY DCU Data Register

Name: DDR3PHY_DCUDR
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

DCU Data Register is used to hold the data to be written to or read from the cache.

Bit	31	30	29	28	27	26	25	24
	CDATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CDATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CDATA[31:0] Cache Data
Data to be written to or read from a cache. This data corresponds to the cache word slice specified by the DCU Address Register.

18.4.38 DDR3PHY DCU Run Register

Name: DDR3PHY_DCURR
Offset: 0xC8
Reset: 0x00000000
Property: Read/Write

DCU Run Register is used to trigger the running of DRAM commands in the DRAM Command Unit command cache.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	XCEN	RCEN	SCOF	SONF	NFAIL[7:4]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFAIL[3:0]				EADDR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SADDR[3:0]				DINST[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 23 – XCEN Expected Compare Enable
Indicates if set that read data coming back from the SDRAM should be should be compared with the expected data.

Bit 22 – RCEN Read Capture Enable
Indicates if set that read data coming back from the SDRAM should be captured into the read data cache.

Bit 21 – SCOF Stop Capture On Full
Specifies if set that the capture of read data should stop when the capture cache is full.

Bit 20 – SONF Stop On Nth Fail
Specifies if set that the execution of commands and the capture of read data should stop when there are N read data failures. The number of failures is specified by NFAIL. Otherwise commands execute until the end of the program or until manually stopped using a STOP command.

Bits 19:12 – NFAIL[7:0] Number of Failures
Specifies the number of failures after which the execution of commands and the capture of read data should stop if SONF bit of this register is set. Execution of commands and the capture of read data will stop after (NFAIL+1) failures if SONF is set.

Bits 11:8 – EADDR[3:0] End Address
Cache word address where the execution of command should end.

Bits 7:4 – SADDR[3:0] Start Address
Cache word address where the execution of commands should begin.

Bits 3:0 – DINST[3:0] DCU Instruction
Selects the DCU command to be executed.

Value	Description
0000	NOP: No operation
0001	Run: Triggers the execution of commands in the command cache
0010	Stop: Stops the execution of commands in the command cache.
0011	Stop Loop: Stops the execution of an infinite loop in the command cache.
0100	Reset: Resets all DCU run time registers
0101 - 1111	Reserved

18.4.39 DDR3PHY DCU Loop Register

Name: DDR3PHY_DCULR
Offset: 0xCC
Reset: 0xF0000000
Property: Read/Write

DCU Loop Register is used to configure the looping when commands are executed from the command cache.

Bit	31	30	29	28	27	26	25	24
	XLEADDR[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	1	1	1	1				

Bit	23	22	21	20	19	18	17	16
							IDA	LINF
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	LCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LEADDR[3:0]				LSADDR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – XLEADDR[3:0] Expected Data Loop End Address
The last expected data cache word address that contains valid expected data. Expected data should looped between 0 and this address.

Bit 17 – IDA Increment DRAM Address
Indicates if set that DRAM addresses should be incremented every time a DRAM read/write command inside the loop is executed.

Bit 16 – LINF Loop Infinite
Indicates if set that the loop should be executed indefinitely until stopped by the STOP command. Otherwise the loop is execute LCNT times.

Bits 15:8 – LCNT[7:0] Loop Count
The number of times that the loop should be executed if LINF is not set.

Bits 7:4 – LEADDR[3:0] Loop End Address
Command cache word address where the loop should end.

Bits 3:0 – LSADDR[3:0] Loop Start Address
Command cache word address where the loop should start.

18.4.40 DDR3PHY DCU General Configuration Register

Name: DDR3PHY_DCUGCR
Offset: 0xD0
Reset: 0x00000000
Property: Read/Write

DCU General Configuration Register is used to specify miscellaneous DCU configurations prior to triggering the execution of commands.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	RCSW[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RCSW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RCSW[15:0] Read Capture Start Word
The capture and compare of read data should start after Nth word. For example setting this value to 12 will skip the first 12 read data.

18.4.41 DDR3PHY DCU Timing Parameter Register

Name: DDR3PHY_DCUTPR
Offset: 0xD4
Reset: 0x00000000
Property: Read/Write

The DCU Timing Parameter Register is used to program generic timing parameters for the repeat field of the command cache.

Bit	31	30	29	28	27	26	25	24
	tDCUT3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	tDCUT2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	tDCUT1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	tDCUT0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – tDCUT3[7:0] DCU Generic Timing Parameter 3

Bits 23:16 – tDCUT2[7:0] DCU Generic Timing Parameter 2

Bits 15:8 – tDCUT1[7:0] DCU Generic Timing Parameter 1

Bits 7:0 – tDCUT0[7:0] DCU Generic Timing Parameter 0

18.4.42 DDR3PHY DCU Status Register 0

Name: DDR3PHY_DCUSR0
Offset: 0xD8
Reset: 0x00000000
Property: Read/Write

DCU status registers are used to report various status of the DCU, including number of reads returned and number of read failures.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CFULL	CFAIL	RDONE
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CFULL Capture Full
Indicates if set that the capture cache is full.

Bit 1 – CFAIL Capture Fail
Indicates if set that at least one read data word has failed.

Bit 0 – RDONE Run Done
Indicates if set that the DCU has finished executing the commands in the command cache. This bit is also set to indicate that a STOP command has successfully been executed and command execution has stopped.

18.4.43 DDR3PHY DCU Status Register 1

Name: DDR3PHY_DCUSR1
Offset: 0xDC
Reset: 0x00000000
Property: Read/Write

DCU status registers are used to report various status of the DCU, including number of reads returned and number of read failures.

Bit	31	30	29	28	27	26	25	24
	LPCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FLCND[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RDCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RDCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – LPCNT[7:0] Loop Count
Indicates the value of the loop count. This is useful when the program has stooped because of failures to assess how many reads were executed before first fail.

Bits 23:16 – FLCND[7:0] Fail Count
Number of read words that have failed.

Bits 15:0 – RDCNT[15:0] Read Count
Number of read words returned from the SDRAM.

18.4.44 DDR3PHY BIST Run Register

Name: DDR3PHY_BISTRR
Offset: 0x100
Reset: 0x00000000
Property: Read/Write

BIST Run Register is used to trigger the running of built-in-self test routines on both the address/command lane and the date byte lanes.

Bit	31	30	29	28	27	26	25	24
							BCKSEL[2:1]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	BCKSEL[0]			BDXSEL[1:0]		BDPAT[1:0]		BDMEN
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BACEN	BDXEN	BSONF	NFAIL[7:3]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFAIL[2:0]			BINF	BMODE	BINST[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:23 – BCKSEL[2:0] BIST CK Select
Selects the CK to be used for capturing loopback data on the address/command lane.

Value	Description
000	CK
001	Reserved
010	Reserved
011	Reserved
100	CK#
101	Reserved
110	Reserved
111	Reserved

Bits 20:19 – BDXSEL[1:0] BIST DATX8 Select
Select the byte lane for comparison of loopback/read data. Valid values are 0 to 8.

Bits 18:17 – BDPAT[1:0] BIST Data Pattern
Selects the data pattern used during BIST.

Value	Description
00	Walking 0
01	Walking 1
10	LFSR-based pseudo-random
11	User programmable

Bit 16 – BDMEN BIST Data Mask Enable
Enables if set that the data mask BIST should be included in the BIST run, i.e. data pattern generated and loopback data compared. This is valid only for Loopback mode.

Bit 15 – BACEN BIST AC Enable
Enables the running of BIST on the address/command lane PHY. This bit is exclusive with BDXEN, i.e. both cannot be set to '1' at the same time.

Bit 14 – BDXEN BIST DATX8 Enable
Enables the running of BIST on the data byte lane PHYs. This bit is exclusive with BACEN, i.e. both cannot be set to '1' at the same time.

Bit 13 – BSONF BIST Stop On Nth Fail
Specifies if set that the BIST should stop when an nth data word or address/command comparison error has been encountered.

Bits 12:5 – NFAIL[7:0] Number of Failures
Specifies the number of failures after which the execution of commands and the capture of read data should stop if BSONF bit of this register is set. Execution of commands and the capture of read data will stop after (NFAIL+1) failures if BSONF is set.

Bit 4 – BINF BIST Infinite Run
Specifies if set that the BIST should be run indefinitely until when it is either stopped or a failure has been encountered. Otherwise BIST is run until number of BIST words specified in the BISTWCR register has been generated.

Bit 3 – BMODE BIST Mode
Selects the mode in which BIST is run.

Value	Name	Description
0	Loopback mode	Address, commands and data loop back at the PHY I/Os.
1	DRAM mode	Address, commands and data go to DRAM for normal memory accesses.

Bits 2:0 – BINST[2:0] BIST Instruction
Selects the BIST instruction to be executed.

Value	Name	Description
000	NOP	No operation
001	Run	Triggers the running of the BIST
010	Stop	Stops the running of the BIST
011	Reset	Resets all BIST run-time registers, such as error counters
100-111	–	Reserved

18.4.45 DDR3PHY BIST Word Count Register

Name: DDR3PHY_BISTWCR
Offset: 0x10C
Reset: 0x00000000
Property: Read/Write

BIST Word Count Register is used to specify the number of words to generate during BIST.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BWCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BWCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BWCNT[15:0] BIST Word Count
Indicates the number of words to generate during BIST. This must be a multiple of DRAM burst length (BL) divided by 2, e.g. for BL=8, valid values are 4, 8, 12, 16, and so on.

18.4.46 DDR3PHY BIST Mask 0 Register

Name: DDR3PHY_BISTMSKR0
Offset: 0x104
Reset: 0x00000000
Property: Read/Write

BIST mask registers are used to disable data comparison on individual data bits of the address/command lane and data byte lane. A bit that is masked will not contribute to the word error counters or bit error counters.

Bit	31	30	29	28	27	26	25	24
				ODTMSK				CSMSK
Access				R/W				R/W
Reset				0				0

Bit	23	22	21	20	19	18	17	16
				CKEMSK	WEMSK	BAMSK[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AMSK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AMSK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – ODTMSK
Mask bit for ODT.

Bit 24 – CSMSK
Mask bit for CS#.

Bit 20 – CKEMSK
Mask bit for CKE.

Bit 19 – WEMSK
Mask bit for the WE#.

Bits 18:16 – BAMSK[2:0]
Mask bit for each of the up to 3 bank address bits.

Bits 15:0 – AMSK[15:0]
Mask bit for each of the up to 16 address bits.

18.4.47 DDR3PHY BIST Mask 1 Register

Name: DDR3PHY_BISTMSKR1
Offset: 0x108
Reset: 0x00000000
Property: Read/Write

BIST mask registers are used to disable data comparison on individual data bits of the address/command lane and data byte lane. A bit that is masked will not contribute to the word error counters or bit error counters.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CASMSK	RASMSK	DMMSK[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DQMSK[1:0]	
Access							R/W	R/W
Reset							0	0

Bit 19 – CASMSK
Mask bit for the CAS.

Bit 18 – RASMSK
Mask bit for the RAS.

Bits 17:16 – DMMSK[1:0]
Mask bit for the data mask (DM) bit.

Bits 1:0 – DQMSK[1:0]
Mask bit for each of the 8 data (DQ) bits.

18.4.48 DDR3PHY BIST LFSR Seed Register

Name: DDR3PHY_BISTLSR
Offset: 0x110
Reset: 0x00000000
Property: Read/Write

BIST LFSR Seed Register specifies the seed for generating pseudo-random BIST patterns.

Bit	31	30	29	28	27	26	25	24
	SEED[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SEED[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SEED[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SEED[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEED[31:0]
LFSR seed for pseudo-random BIST patterns.

18.4.49 DDR3PHY BIST Address 0 Register

Name: DDR3PHY_BISTAR0
Offset: 0x114
Reset: 0x00000000
Property: Read/Write

BIST address registers specify the starting address and address increment used during BIST when in DRAM mode. The BIST address is based on a decoded bank/row/column address. The user should ensure the column address starts at the beginning of a burst boundary. For example, for BL8, A[2:0] = DTCOL[2:0] = 0.

Bit	31	30	29	28	27	26	25	24
		BBANK[2:0]			BROW[15:12]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BROW[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BROW[3:0]				BCOL[11:8]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:28 – BBANK[2:0] BIST Bank Address
Selects the SDRAM bank address to be used during BIST.

Bits 27:12 – BROW[15:0] BIST Row Address
Selects the SDRAM row address to be used during BIST.

Bits 11:0 – BCOL[11:0] BIST Column Address
Selects the SDRAM column address to be used during BIST. The lower bits of this address must be “0000” for BL16, “000” for BL8, “00” for BL4 and “0” for BL2.

18.4.50 DDR3PHY BIST Address 1 Register

Name: DDR3PHY_BISTAR1
Offset: 0x118
Reset: 0x00000000
Property: Read/Write

BIST address registers specify the starting address and address increment used during BIST when in DRAM mode. The BIST address is based on a decoded bank/row/column address. The user should ensure the column address starts at the beginning of a burst boundary. For example, for BL8, A[2:0] = DTCOL[2:0] = 0.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BAINC[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAINC[3:0]				BMRANK[1:0]		BRANK[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:4 – BAINC[11:0] BIST Address Increment
Selects the value by which the SDRAM address is incremented for each write/read access. This value must be at the beginning of a burst boundary, i.e. the lower bits must be “0000” for BL16, “000” for BL8, “00” for BL4 and “0” for BL2.

Bits 3:2 – BMRANK[1:0] BIST Maximum Rank
Maximum SDRAM rank to be used during BIST. Write to 0 (only one rank).

Bits 1:0 – BRANK[1:0] BIST Rank
SDRAM rank to be used during BIST. Write to 0 (only one rank).

18.4.51 DDR3PHY BIST Address 2 Register

Name: DDR3PHY_BISTAR2
Offset: 0x11C
Reset: 0x00000000
Property: Read/Write

BIST address registers specify the starting address and address increment used during BIST when in DRAM mode. The BIST address is based on a decoded bank/row/column address. The user should ensure the column address starts at the beginning of a burst boundary. For example, for BL8, A[2:0] = DTCOL[2:0] = 0.

Bit	31	30	29	28	27	26	25	24
		BMBANK[2:0]			BMROW[15:12]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BMROW[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BMROW[3:0]				BMCOL[11:8]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BMCOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:28 – BMBANK[2:0] BIST Maximum Bank Address
Specifies the maximum SDRAM bank address to be used during BIST before the address increments to the next rank.

Bits 27:12 – BMROW[15:0] BIST Maximum Row Address
Specifies the maximum SDRAM row address to be used during BIST before the address increments to the next bank.

Bits 11:0 – BMCOL[11:0] BIST Maximum Column Address
Specifies the maximum SDRAM column address to be used during BIST before the address increments to the next row.

18.4.52 DDR3PHY BIST User Data Pattern Register

Name: DDR3PHY_BISTUDPR
Offset: 0x120
Reset: 0x00000000
Property: Read/Write

This register specifies the user data pattern to be used during BIST. The pattern transmitted on DQ[2*n] is BUDP0[0] on the first beat, BUDP0[1] on the second beat and so on. BUDP1 is transmitted on DQ[2*n+1] in a similar way. After the 16th beat, the data pattern wraps around and is repeated until BIST is stopped.

Bit	31	30	29	28	27	26	25	24
	BUDP1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUDP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BUDP0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUDP0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – BUDP1[15:0] BIST User Data Pattern 1
Data to be applied on odd DQ pins during BIST.

Bits 15:0 – BUDP0[15:0] BIST User Data Pattern 0
Data to be applied on even DQ pins during BIST.

18.4.53 DDR3PHY General Status Register

Name: DDR3PHY_BISTGSR
Offset: 0x124
Reset: 0x00000000
Property: Read/Write

BIST General Status Register is used to return miscellaneous status of the BIST engine, including done status.

Bit	31	30	29	28	27	26	25	24
	CASBER[1:0]		RASBER[1:0]		DMBER[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BDXERR	BACERR	BDONE
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:30 – CASBER[1:0] CAS Bit Error
Indicates the number of bit errors on CAS.

Bits 29:28 – RASBER[1:0] RAS Bit Error
Indicates the number of bit errors on RAS.

Bits 27:24 – DMBER[3:0] DM Bit Error
Indicates the number of bit errors on data mask (DM) bit. DMBER[1:0] are for the first DM beat, and DMBER[3:2] are for the second DM beat.

Bit 2 – BDXERR BIST Data Error
Indicates if set that there is a data comparison error in the byte lane.

Bit 1 – BACERR BIST Address/Command Error
Indicates if set that there is a data comparison error in the address/command lane.

Bit 0 – BDONE BIST Done
Indicates if set that the BIST has finished executing. This bit is reset to zero when BIST is triggered.

18.4.54 DDR3PHY Word Error Register

Name: DDR3PHY_BISTWER
Offset: 0x128
Reset: 0x00000000
Property: Read/Write

BIST Word Error Register is used to return the number of errors on the full word (all bits) of the address/command lane and data byte lane. An error on any bit of the address/command bus will increment the address/command word error count. Similarly any error on the data bits or data mask bits will increment the data byte lane word error count.

Bit	31	30	29	28	27	26	25	24
	DXWER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DXWER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ACWER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACWER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DXWER[15:0] Byte Word Error
Indicates the number of word errors on the byte lane. An error on any bit of the data bus including the data mask bit increments the error count.

Bits 15:0 – ACWER[15:0] Address/Command Word Error
Indicates the number of word errors on the address/command lane. An error on any bit of the address/command bus increments the error count.

18.4.55 DDR3PHY BIST Bit Error 0 Register

Name: DDR3PHY_BISTBER0
Offset: 0x12C
Reset: 0x00000000
Property: Read/Write

BIST error registers are used to return the number of errors on individual data bits of the address/command lane and data byte lane. Bit error registers only return up to 4 errors on the address/command bits and up to 8 errors on the data bits. To diagnose more errors on a bit, use the BIST mask registers to mask all but the bit of interest and re-run the BIST. In this case the word error counter is equal to the bit error counter.

Bit	31	30	29	28	27	26	25	24
	ABER[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ABER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ABER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ABER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ABER[31:0] Address Bit Error
Each group of two bits indicates the bit error count on each of the up to 16 address bits. [1:0] is the error count for A[0], [3:2] for A[1], and so on.

18.4.56 DDR3PHY BIST Bit Error 1 Register

Name: DDR3PHY_BISTBER1
Offset: 0x130
Reset: 0x00000000
Property: Read/Write

BIST error registers are used to return the number of errors on individual data bits of the address/command lane and data byte lane. Bit error registers only return up to 4 errors on the address/command bits and up to 8 errors on the data bits. To diagnose more errors on a bit, use the BIST mask registers to mask all but the bit of interest and re-run the BIST. In this case the word error counter is equal to the bit error counter.

Bit	31	30	29	28	27	26	25	24
							ODTBER[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							CSBER[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							CKEBER[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	WEBER[1:0]		BABER[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:24 – ODTBER[1:0] ODT Bit Error
Bit error count on ODT bits.

Bits 17:16 – CSBER[1:0] CS# Bit Error
Bit error count on CS# bits.

Bits 9:8 – CKEBER[1:0] CKE Bit Error
Bit error count on CKE bits.

Bits 7:6 – WEBER[1:0] WE# Bit Error
Bit error count on WE#.

Bits 5:0 – BABER[5:0] Bank Address Bit Error
Each group of two bits indicates the bit error count on each of the up to 3 bank address bits. [1:0] is the error count for BA[0], [3:2] for BA[1], and so on.

18.4.57 DDR3PHY BIST Bit Error 2 Register

Name: DDR3PHY_BISTBER2
Offset: 0x134
Reset: 0x00000000
Property: Read/Write

BIST error registers are used to return the number of errors on individual data bits of the address/command lane and data byte lane. Bit error registers only return up to 4 errors on the address/command bits and up to 8 errors on the data bits. To diagnose more errors on a bit, use the BIST mask registers to mask all but the bit of interest and re-run the BIST. In this case the word error counter is equal to the bit error counter.

Bit	31	30	29	28	27	26	25	24
	DQBER[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DQBER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DQBER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DQBER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DQBER[31:0] Data Bit Error
The first 16 bits indicate the error count for the first data beat (i.e. the data driven out on DQ[7:0] on the rising edge of DQS). The second 16 bits indicate the error on the second data beat (i.e. the error count of the data driven out on DQ[7:0] on the falling edge of DQS). For each of the 16-bit group, the first 2 bits are for DQ[0], the second for DQ[1], and so on.

18.4.58 DDR3PHY BIST Word Count Status Register

Name: DDR3PHY_BISTWCSR
Offset: 0x138
Reset: 0x00000000
Property: Read/Write

BIST Word Count Status Register is used to return the number of words received from the address/command lane and data byte lane.

Bit	31	30	29	28	27	26	25	24
	DXWCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DXWCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ACWCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACWCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DXWCNT[15:0] Byte Word Count
Indicates the number of words received from the byte lane.

Bits 15:0 – ACWCNT[15:0] Address/Command Word Count
Indicates the number of words received from the address/command lane.

18.4.59 DDR3PHY Fail Word 0 Register

Name: DDR3PHY_BISTFWR0
Offset: 0x13C
Reset: 0x00000000
Property: Read/Write

BIST fail word registers are used to return the last failing word of the address/command lane and data byte lane.

Bit	31	30	29	28	27	26	25	24
				ODTWEBS				CSWEBS
Access				R/W				R/W
Reset				0				0

Bit	23	22	21	20	19	18	17	16
				CKEWEBS	WEWEBS	BAWEBS[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AWEBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AWEBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – ODTWEBS
Bit status during a word error for ODT bits.

Bit 24 – CSWEBS
Bit status during a word error for CS# bits.

Bit 20 – CKEWEBS
Bit status during a word error for CKE bits.

Bit 19 – WEWEBS
Bit status during a word error for the WE#.

Bits 18:16 – BAWEBS[2:0]
Bit status during a word error for each of the up to 3 bank address bits.

Bits 15:0 – AWEBS[15:0]
Bit status during a word error for each of the up to 16 address bits.

18.4.60 DDR3PHY Fail Word 1 Register

Name: DDR3PHY_BISTFWR1
Offset: 0x140
Reset: 0x00000000
Property: Read/Write

BIST fail word registers are used to return the last failing word of the address/command lane and data byte lane.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CASWEBS	RASWEBS	DMWEBS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DQWEBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DQWEBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 19 – CASWEBS
Bit status during a word error for the CAS.

Bit 18 – RASWEBS
Bit status during a word error for the RAS.

Bits 17:16 – DMWEBS[1:0]
Bit status during a word error for the data mask (DM) bit. DMWEBS [0] is for the first DM beat, and DMWEBS [1] is for the second DM beat.

Bits 15:0 – DQWEBS[15:0]
Bit status during a word error for each of the 8 data (DQ) bits. The first 8 bits indicate the status of the first data beat (i.e. the status of the data driven out on DQ[7:0] on the rising edge of DQS). The second 8 bits indicate the status of the second data beat (i.e. the status of the data driven out on DQ[7:0] on the falling edge of DQS). For each of the 8-bit group, the first bit is for DQ[0], the second bit is for DQ[1], and so on.

18.4.61 DDR3PHY ZQ Impedance Control Register 0

Name: DDR3PHY_ZQ0CR0
Offset: 0x180
Reset: 0x00000000
Property: Read/Write

These registers are used for impedance control and calibration to enable the programmable and PVT-compensated on-die termination (ODT) and output impedance of the functional SSTL cells.

Bit	31	30	29	28	27	26	25	24
	ZQPD	ZCAL	ZCALBYP	ZDEN	ZDATA[27:24]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ZDATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ZDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ZQPD ZQ Power Down
Powers down, if set, the PZQ cell. Not applicable to D3A I/Os.

Bit 30 – ZCAL Impedance Calibration Trigger
A write of '1' to this bit triggers impedance calibration to be performed by the impedance control logic. The impedance calibration trigger bit is self-clearing and returns back to '0' when the calibration is complete. Note: If ZDEN is set, then the ZCAL bit should be set to 0.

Bit 29 – ZCALBYP Impedance Calibration Bypass
Disables, if set, impedance calibration of this ZQ control block when impedance calibration is triggered globally using the ZCAL bit of PIR. Impedance calibration of this ZQ block may be triggered manually using ZCAL.

Bit 28 – ZDEN Impedance Over-Ride Enable
When this bit is set, it allows users to directly drive the impedance control using the data programmed in the ZQDATA field. Otherwise, the control is generated automatically by the impedance control logic. Note: If ZDEN is set, then the ZCAL bit should be set to 0.

Bits 27:0 – ZDATA[27:0] Impedance Over-Ride Data
Data used to directly drive the impedance control. Field mapping is as follows:

[27:20]	Reserved, returns zeros on reads
[19:15]	Pull-up on-die termination impedance select
[14:10]	Pull-down on-die termination impedance select
[9:5]	Pull-up output impedance select
[4:0]	Pull-down output impedance select

18.4.62 DDR3PHY ZQ Impedance Control Register 1

Name: DDR3PHY_ZQ0CR1
Offset: 0x184
Reset: 0x00000000
Property: Read/Write

These registers are used for impedance control and calibration to enable the programmable and PVT-compensated on-die termination (ODT) and output impedance of the functional SSTL cells.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ZPROG[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ZPROG[7:0] Impedance Divide Ratio
Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination.

ZPROG[3:0] (decimal)	Output impedance divide select (Ohms)
5	80
7	60
9	48
11	40
13	34
ZPROG[7:4] (decimal)	On-die termination divide select (Ohms)
1	120
4	75
5	60
6	50
8	40

18.4.63 DDR3PHY ZQ Status Register 0

Name: DDR3PHY_ZQ0SR0
Offset: 0x188
Reset: 0x00000000
Property: Read/Write

These registers are used to provide the status for impedance control and calibration to enable the programmable and PVT-compensated on-die termination (ODT) and output impedance of the functional SSTL cells.

Bit	31	30	29	28	27	26	25	24
	ZDONE	ZERR			ZCTRL[27:24]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ZCTRL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ZCTRL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ZCTRL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ZDONE Impedance Calibration Done
Indicates that impedance calibration has completed.

Bit 30 – ZERR Impedance Calibration Error
If set, indicates that there was an error during impedance calibration.

Bits 27:0 – ZCTRL[27:0] Impedance Control
Current value of impedance control. Field mapping is as follows:

[27:20]	Reserved, returns zeros on reads
[19:15]	Pull-up on-die termination impedance select
[14:10]	Pull-down on-die termination impedance select
[9:5]	Pull-up output impedance select
[4:0]	Pull-down output impedance select

18.4.64 DDR3PHY ZQ Status Register 1

Name: DDR3PHY_ZQ0SR1
Offset: 0x18C
Reset: 0x00000000
Property: Read/Write

These registers are used to provide the status for impedance control and calibration to enable the programmable and PVT-compensated on-die termination (ODT) and output impedance of the functional SSTL cells.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	OPU[1:0]		OPD[1:0]		ZPU[1:0]		ZPD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:6 – OPU[1:0] On-Die Termination (ODT) Pull-Up Calibration Status
Similar status encodings as ZPD.

Bits 5:4 – OPD[1:0] On-Die Termination (ODT) Pull-Down Calibration Status
Similar status encodings as ZPD.

Bits 3:2 – ZPU[1:0] Output Impedance Pull-Up Calibration Status
Similar status encodings as ZPD.

Bits 1:0 – ZPD[1:0] Output Impedance Pull-Down Calibration Status
Valid status encodings are:

00	Completed with no errors
01	Overflow error
10	Underflow error
11	Calibration in progress

18.4.65 DDR3PHY DATX8 General Configuration Register

Name: DDR3PHY_DXxGCR
Offset: 0x01C0 + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register is used for miscellaneous configurations specific to a particular instantiation of the DATX8 macro. For D3A I/Os, the IOM register bits connect to the I/O IDDQ pins. Furthermore, the I/O power-down register bits (PDD, and PDR, and PD) fields have no relevance for D3A I/Os.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								R0RVSL[2]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	R0RVSL[1:0]		RTTOAL	RTTOH[1:0]		DQRTT	DQSRTT	DSEN[1]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DSEN[0]	DQSRPD	DXPDR	DXPDD	DXIOM	DQODT	DQSODT	DXEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16:14 – R0RVSL[2:0] Rank 0 ITMD Read Valid System Latency
Used to specify the read valid system latency relative to the ideal placement of the ITMD read valid signal when DXCCR.RVSEL is set to 0. Power-up default is 011 (i.e. ideal placement of the read valid signal). The RVSL field is initially set by the PUB during automatic read valid training but this value can be overwritten by a direct write to this register.

Value	Description
000	Read valid system latency = ideal placement - 3
001	Read valid system latency = ideal placement - 2
010	Read valid system latency = ideal placement - 1
011	Read valid system latency = ideal placement
100	Read valid system latency = ideal placement + 1
101	Read valid system latency = ideal placement + 2
110	Read valid system latency = ideal placement +3
111	Reserved

Bit 13 – RTTOAL RTT On Additive Latency
Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles.

Value	Description
0	ODT control is set to DQSODT/DQODT almost two cycles before read data preamble.
1	ODT control is set to DQSODT/DQODT almost one cycle before read data preamble.

Bits 12:11 – RTTOH[1:0] RTT Output Hold
Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0') when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble.

Bit 10 – DQRTT DQ Dynamic RTT Control
Indicates, if set, that the ODT control of DQ/DM SSTL I/Os be dynamically controlled by setting it to the value in DQODT during reads and disabling it (setting it to '0') during any other cycle. If this bit is not set, then the ODT control of DQ SSTL I/Os is always set to the value in DQODT. Since dynamic ODT is on by default, when using LPDDR2/LPDDR3 this bit must be set to 0 since LPDDR2/LPDDR3 does not require ODT to be on.

Bit 9 – DQSRTT DQS Dynamic RTT Control
Indicates, if set, that the ODT control of DQS SSTL I/Os be dynamically controlled by setting it to the value in DQSODT during reads and disabling it (setting it to '0') during any other cycle. If this bit is not set, then the ODT control of DQS SSTL I/Os is always set to the value in DQSODT field. Since dynamic ODT is on by default, when using LPDDR2/LPDDR3 this bit must be set to 0 since LPDDR2/LPDDR3 does not require ODT to be on.

Bits 8:7 – DSEN[1:0] Write DQS Enable
Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated.

Value	Description
00	DQS disabled (driven to constant 0)
01	DQS toggling with normal polarity (this should be the default setting)
10	DQS toggling with inverted polarity
11	DQS disabled (driven to constant 1)

Bit 6 – DQSRPD DQSR Power Down
Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit (see 18.4.10. DDR3PHY_DXCCR).

Bit 5 – DXPDR Data Power Down Receiver
Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit (see 18.4.10. DDR3PHY_DXCCR).

Bit 4 – DXPDD Data Power Down Driver
Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit (see 18.4.10. DDR3PHY_DXCCR).

Bit 3 – DXIOM Data I/O Mode
Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8 (see 18.4.10. DDR3PHY_DXCCR).

Bit 2 – DQODT Data On-Die Termination
Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit (see 18.4.10. DDR3PHY_DXCCR).

Bit 1 – DQSODT DQS On-Die Termination
Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit (see 18.4.10. DDR3PHY_DXCCR).

Bit 0 – DXEN Data Byte Enable
Enables, if set, the DATX8 and SSTL I/Os used on the data byte. Setting this bit to '0' disables the byte, i.e. the byte SSTL I/Os are put in Power-Down mode and the DLL in the DATX8 is put in Bypass mode. After changing a Byte Lane from disabled to enabled, the DLL for that Byte Lane must be reset and re-locked. Software can use the PIR registers DLLSRST and DLLLOCK to accomplish this (reset and re-locks all DLLs in the DDR PHY).

18.4.66 DDR3PHY DATX8 General Status Register 0

Name: DDR3PHY_DXxGSR0
Offset: 0x01C4 + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This is a general status register for the DATX8. They indicate among other things DQS drift and data training status.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DTPASS[2:0]							DTIERR
Access	R/W	R/W	R/W					R/W
Reset	0	0	0					0

Bit	7	6	5	4	3	2	1	0
				DTERR				DTDONE
Access				R/W				R/W
Reset				0				0

Bits 15:13 – DTPASS[2:0] DQS GateTraining Pass Count
The number of passing configurations during DQS gate training.

Bit 8 – DTIERR DQS Gate Training Intermittent Error
If set, indicates that there was an intermittent error during DQS gate training of the byte, such as a pass was followed by a fail then followed by another pass.

Bit 4 – DTERR DQS Gate Training Error
If set, indicates that a valid DQS gating window could not be found during DQS gate training of the byte.

Bit 0 – DTDONE Data Training Done
Indicates, if set, that the byte has finished doing data training.

18.4.67 DDR3PHY DATX8 General Status Register 1

Name: DDR3PHY_DXxGSR1
Offset: 0x01C8 + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This is a general status register for the DATX8. They indicate among other things DQS drift and data training status.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RVPASS[1:0]					RVIERR
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	15	14	13	12	11	10	9	8
				RVERR				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
			DQSDFT[1:0]					DFTERR
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 21:20 – RVPASS[1:0] Read Valid Training Pass Count
The number of passing configurations during read valid training.

Bit 16 – RVIERR Read Valid Training Intermittent Error
If set, indicates that there was an intermittent error during read valid training of the byte, such as a pass was followed by a fail then followed by another pass.

Bit 12 – RVERR Read Valid Training Error
If set, indicates that a valid read valid placement could not be found during read valid training of the byte.

Bits 5:4 – DQSDFT[1:0] DQS Drift
Used to report the drift on the read data strobe of the data byte.

Value	Description
`00`	No drift
`01`	90° drift
`10`	180° drift
`11`	270° drift or more

Bit 0 – DFTERR DQS Drift Error
If set, indicates that the byte read data strobe has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR).

18.4.68 DDR3PHY DATX8 DLL Control Register

Name: DDR3PHY_DXxDLLCR
Offset: 0x01CC + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

DATX8 DLL Control Register is used to control the DLL on the byte lane.

Bit	31	30	29	28	27	26	25	24
	DLLDIS	DLLSRST						
Access	R/W	R/W						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
					SDLBMODE	ATESTEN	SDPHASE[3:2]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDPHASE[1:0]		SSTART[1:0]		MFWDLY[2:0]			MFBDLY[2]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MFBDLY[1:0]		SFWDLY[2:0]			SFBDLY[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – DLLDIS DLL Disable
A disabled DLL is bypassed. Default ('0') is DLL enabled.

Bit 30 – DLLSRST DLL Soft Reset
Soft resets the byte DLL by driving the DLL soft reset pin.

Bit 19 – SDLBMODE Client DLL Loopback Mode
If this bit is set, the client DLL is put in Loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in Loopback mode such as during BIST loopback.

Bit 18 – ATESTEN Analog Test Enable
Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.

Bits 17:14 – SDPHASE[3:0] Client DLL Phase Trim
Selects the phase difference between the input clock and the corresponding output clock of the client DLL.

Value	Description
`0000`	90
`0001`	72
`0010`	54
`0011`	36
`0100`	108
`0101`	90
`0110`	72
`0111`	54
`1000`	126
`1001`	108
`1010`	90

Value	Description
1011	72
1100	144
1101	126
1110	108
1111	90

Bits 13:12 – SSTART[1:0] Client Auto Start-Up
Used to control how the client DLL starts up relative to the host DLL locking.

Value	Description
0X	Client DLL automatically starts up once the host DLL has achieved lock.
10	The automatic startup of the client DLL is disabled; the phase detector is disabled.
11	The automatic startup of the client DLL is disabled; the phase detector is enabled.

Bits 11:9 – MFWDLY[2:0] Host Feed-Forward Delay Trim
Used to trim the delay in the host DLL feed-forward path.

Value	Description
000	Minimum delay
111	Maximum delay

Bits 8:6 – MFBDLY[2:0] Host Feed-Back Delay Trim
Used to trim the delay in the host DLL feedback path.

Value	Description
000	Minimum delay
111	Maximum delay

Bits 5:3 – SFWDLY[2:0] Client Feed-Forward Delay Trim
Used to trim the delay in the client DLL feed-forward path.

Value	Description
000	Minimum delay
111	Maximum delay

Bits 2:0 – SFBDLY[2:0] Client Feed-Back Delay Trim
Used to trim the delay in the client DLL feedback path.

Value	Description
000	Minimum delay
111	Maximum delay

18.4.69 DDR3PHY DATX8 DQ Timing Register

Name: DDR3PHY_DXxDQTR
Offset: 0x01D0 + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register is used to tune the delays within the ITMs on each data bit to maximize the data eye during reads from the SDRAM. The default settings for this register ensure the delay on each data bit is matched to the delay of the data strobe through the client DLL and clock tree. These settings provide optimal alignment for these signal inputs on the ITMD cell. The DQTR register allows this delay to be slightly decreased.

The PHY is designed such that the default delay setting results in the optimal timing. Users may attempt to fine-tune the DQ/DQS relationship further by modifying these registers. If users elect to modify these registers, use of a modified data training sequence is suggested to find the optimum setting.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DQDLY1[3:0]				DQDLY0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7 – DQDLYx DQ Delay
Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the client DLL and clock tree. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1].

Value	Description
`00`	Nominal delay
`01`	Nominal delay + 1 step
`10`	Nominal delay + 2 steps
`11`	Nominal delay + 3 steps

18.4.70 DDR3PHY DATX8 DQS Timing Register

Name: DDR3PHY_DXxDQSTR
Offset: 0x01D4 + x*0x40 [x=0..1]
Reset: 0x00000000
Property: Read/Write

This register is used to control the timing of the DQS gating. It is also used to tune the delays within the ITMs on each DQS/DQS_b strobe to maximize the data eye during reads from the SDRAM. Usually, the delay on each data strobe is matched to the delay of the data bit through the client DLL and clock tree. The DQSTR register allows this delay to be slightly increased or decreased.

The PHY is designed such that the nominal delay setting results in the optimal timing. Users may attempt to fine-tune the DQ/DQS relationship further by modifying these registers. If users elect to modify these registers, use of a modified data training sequence is suggested to find the optimum setting.

Bit	31	30	29	28	27	26	25	24
							DQSNDLY[2:1]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	DQSNDLY[0]	DQSDLY[2:0]						
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
			R0DGPS[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit	7	6	5	4	3	2	1	0
						R0DGSL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 25:23 – DQSNDLY[2:0] DQS# Delay
Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the client DLL and clock tree. DQSDLY controls the delay on DQS strobe and DQSNDLY controls the delay on DQS#.
Note: After changing this value, an ITM soft reset (PIR.ITMSRST=1, plus PIR.INIT=1) must be issued.

Value	Description
000	Nominal delay - 3 steps
001	Nominal delay - 2 steps
010	Nominal delay - 1 step
011	Nominal delay
100	Nominal delay + 1 step
101	Nominal delay + 2 steps
110	Nominal delay + 3 steps
111	Nominal delay + 4 steps

Bits 25:23 – DMDLY[2:0] DM Delay
Used to adjust the delay of the data mask relative to the nominal delay that is matched to the delay of the data strobes through the client DLL and clock tree. The lower two bits of the DQMDLY controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b.

Value	Description
00	Nominal delay

Value	Description
01	Nominal delay + 1 step
10	Nominal delay + 2 steps
11	Nominal delay + 3 steps

Bits 22:20 – DQSDLY[2:0] DQS Delay
Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the client DLL and clock tree. DQSDLY controls the delay on DQS strobe and DQSNDLY controls the delay on DQS#.
Note: After changing this value, an ITM soft reset (PIR.ITMSRST=1, plus PIR.INIT=1) must be issued.

Value	Description
000	Nominal delay - 3 steps
001	Nominal delay - 2 steps
010	Nominal delay - 1 step
011	Nominal delay
100	Nominal delay + 1 step
101	Nominal delay + 2 steps
110	Nominal delay + 3 steps
111	Nominal delay + 4 steps

Bits 13:12 – R0DGPS[1:0] Rank 0 DQS Gating Phase Select
Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The R0DGPS field is initially set by the PUBL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, this value can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using DDR3PHY_PGCR.

Value	Description
00	90° clock (clk90)
01	180° clock (clk180)
10	270° clock (clk270)
11	360° clock (clk0)

Bits 2:0 – R0DGSL[2:0] Rank 0 DQS Gating System Latency
Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL field is initially set by the PUBL during automatic DQS data training but this value can be overwritten by a direct write to this register.

Value	Description
000	No extra clock cycles
001	1 extra clock cycle
010	2 extra clock cycles
011	3 extra clock cycles
100	4 extra clock cycles
101	5 extra clock cycles
110	Reserved
111	Reserved

19. Boot Strategies

19.1 Standard Boot Strategies

19.1.1 Overview

By default, the chip starts in Standard Boot mode. See Secure Boot Strategies for details on Secure Boot mode enabling.

The system always boots from the ROM memory at address 0x0.

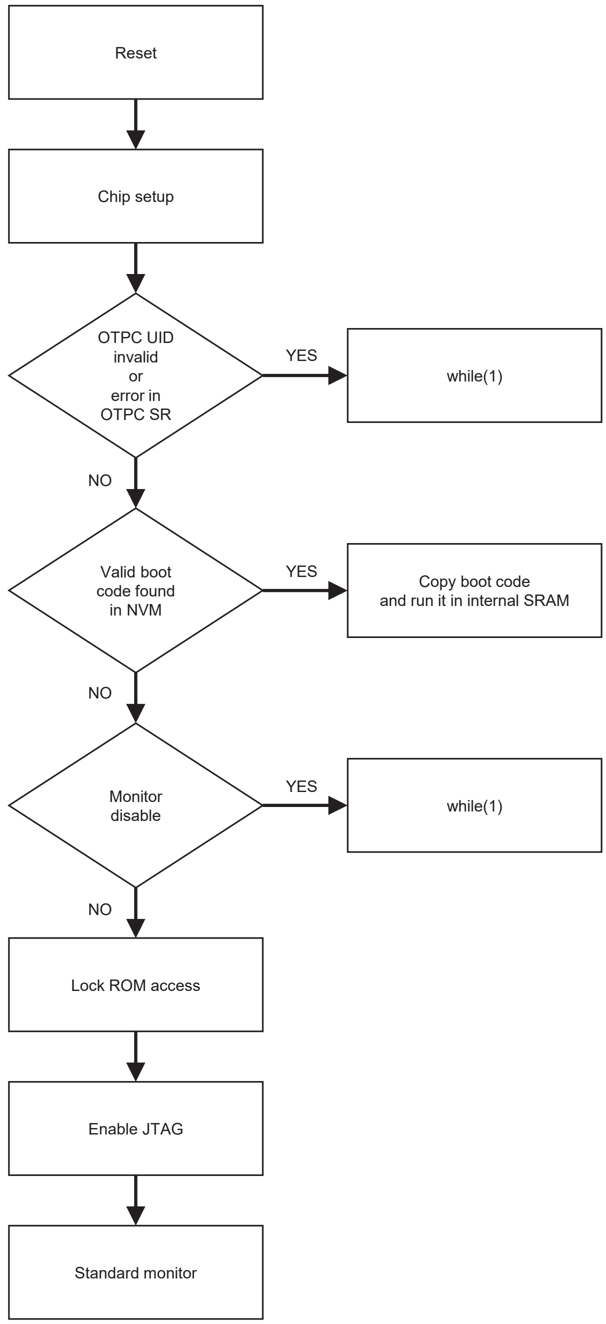
The ROM code is a boot program contained in the embedded ROM. It is also called “Bootloader”.

Note: JTAG access is disabled during execution of the ROM code. It is enabled when jumping into SRAM when a valid code has been found on an external Non-Volatile Memory (NVM) at the same time the ROM memory access is locked. If no valid bootstrap is found on an external NVM, the ROM code enables the USB connection and one UART serial port, starts the standard monitor, locks access to the ROM memory and enables the JTAG connection.

19.1.2 Flow Diagram

The following figure shows the ROM code global flow.

Figure 19-1. ROM Code Flow Diagram



19.1.3 Chip Setup

When the chip is powered on, the processor clock (CPU_CLK) and the main system bus clock (MCK0) source is the main clock (MAINCK), which is fed by the main RC oscillator.

The ROM code performs a low-level initialization that follows the steps described below:

1. SYSPLL is initialized.

2. MCK1 is initialized and sourced by SYSPLL.
3. MCK4 is initialized and sourced by SYSPLL.
4. CPUPLL is initialized.
5. When the CPUPLL is stabilized, the main system bus clock (MCK0) source is switched from the main clock (MAINCK) to the CPUPLL clock. The CPU_CLK frequency is the same as the CPUPLL clock, whereas the MCK0 frequency is the quarter of the CPUPLL clock.

For clock frequencies, see the table Clock Frequencies During External Memory Boot Sequence.

Note: No external crystal or clock is needed during the external boot memories sequence. An external clock source is checked before the launch of the monitor to get a more accurate clock signal for USB.

19.1.4 Standard Boot Configuration

The boot sequence is controlled using the Boot Configuration Packet, stored in the OTP User area and configured through the OTP Controller (refer to OTP Memory Controller (OTPC)).

19.1.4.1 Default Boot Sequence

When no Boot Configuration Packet is available in the OTP User area, the ROM code uses the following settings:

- FLEXCOM3 IOSET 5 is used as a console.
- Boot from SDMMC1 IOSET1 (uses card detect pin)

If no valid code is found in the SD Card or e.MMC, the ROM code goes to the standard monitor.

See the table PIO Driven during Boot Program Execution for further details about PIO settings.

19.1.4.2 Boot Configuration Packet

The boot configuration data are stored in the Boot Configuration Packet in the OTP User area. These data can be used for various boot sequence customizations:

- Enable and configure the boot from selected memories
- Configure the UART port used as a console
- Disable monitor

See Boot Configuration User Interface for a detailed description of all the fields in these data.

Out of factory, the OTP memory is empty. No Boot Configuration Packet is written. During the prototyping phase, the Boot Configuration Packet can be created and stored in a backed up OTP emulation SRAM to avoid an OTP permanent write. To enable this feature, the bit BSC_CR.EMUL_EN must be set. Note that once a real packet has been written in OTP User Area, the Emulation mode cannot be used anymore.

The current running mode of the OTP can be observed by reading BSC_CR.EMUL_EN. If this bit is set, the OTP Emulation mode is enabled, otherwise it is disabled.

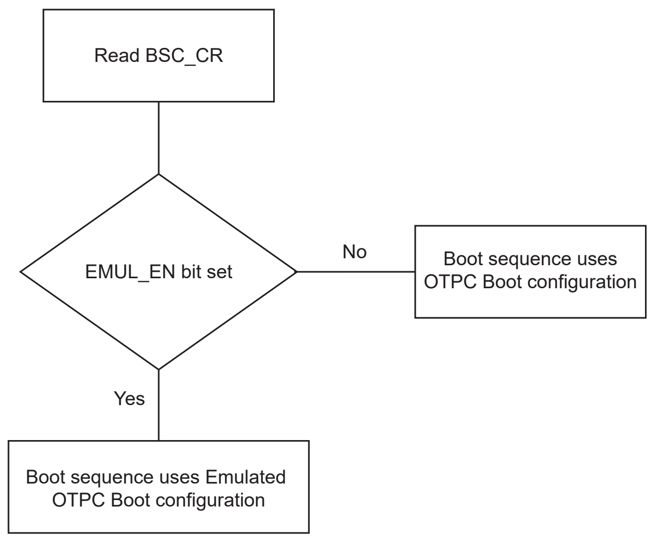
After a reset, the ROM code reads the Boot Configuration Packet from the OTP emulation SRAM if the bit BSC_CR.EMUL_EN is set to 1. Otherwise, the packet is read from the OTP matrix.

The OTP Emulation mode enables the user to test several boot configuration options, including Secure Boot Mode (see Secure Boot Strategies), without programming permanently the OTP.

Note: The OTP emulation SRAM is backed up. So, if the VDDBU power is supplied, the boot configuration is kept when the chip in not powered on.

⚠WARNING The Boot Configuration Packet must be the very first packet to be written in OTP (not in Emulation mode).

Figure 19-2. Boot Configuration Loading



19.1.4.3 Boot Sequence Controller Configuration Register

Name: BSC_CR
Address: 0xE001D054

Bit	31	30	29	28	27	26	25	24
	WPKEY[15:8]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPKEY[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								EMUL_EN
Access								
Reset								

Bits 31:16 – WPKEY[15:0] Write Protect Key

Value	Name	Description
0x6683	PASSWD	Writing any other value in this field aborts the write operation of the BOOT field. Always reads as 0.

Bit 0 – EMUL_EN Emulation Enable

Value	Description
0	Emulation mode is disabled.
1	The OTP user area is emulated in internal SRAM1.

19.1.4.4 Boot Configuration User Interface

Table 19-1. Boot Configuration Packet

Offset	Name	Description
0x00	MON_DIS	Disables monitor
0x04	RESERVED	Reserved
0x08	CONSOLE_PIN	Console pin muxing
0x0C-0x44	MEM_CFGx[2]	Two 32-bit words used to set memory configurations for x=0..6 MEM_CFGx[0]: this word contains mandatory options for all connected memories. MEM_CFGx[1]: this word contains mandatory options for specific memories.

19.1.4.5 Monitor Disable

Name: MON_DIS

Bit	31	30	29	28	27	26	25	24
	DISABLE_MONITOR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DISABLE_MONITOR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DISABLE_MONITOR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DISABLE_MONITOR[7:0]							
Access								
Reset								

Bits 31:0 – DISABLE_MONITOR[31:0] Monitor Disable

Value	Description
`0`	If no boot file is found, launches the monitor.
`Non-Zero`	The monitor is never launched.

19.1.4.6 Console Pin Muxing

Name: CONSOLE_PIN

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	CONSOLE_IOSET[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CONSOLE_IOSET[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CONSOLE_IOSET[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CONSOLE_IOSET[7:0]							
Access								
Reset								

Bits 31:0 – CONSOLE_IOSET[31:0] Selects the pins and UART interface used as a console terminal

Value	Name	Description
0	UART0 IO1	Uses FLEXCOM0 IOSET 1 USART pins
1	UART0 IO2	Uses FLEXCOM0 IOSET 2 USART pins
2	UART0 IO3	Uses FLEXCOM0 IOSET 3 USART pins
3	UART0 IO4	Uses FLEXCOM0 IOSET 4 USART pins
4	UART1 IO1	Uses FLEXCOM1 IOSET 1 USART pins
5	UART1 IO2	Uses FLEXCOM1 IOSET 2 USART pins
6	UART1 IO3	Uses FLEXCOM1 IOSET 3 USART pins
7	UART1 IO4	Uses FLEXCOM1 IOSET 4 USART pins
8	UART2 IO1	Uses FLEXCOM2 IOSET 1 USART pins
9	UART2 IO2	Uses FLEXCOM2 IOSET 2 USART pins
10	UART2 IO3	Uses FLEXCOM2 IOSET 3 USART pins
11	UART2 IO4	Uses FLEXCOM2 IOSET 4 USART pins
12	UART2 IO5	Uses FLEXCOM2 IOSET 5 USART pins
13	UART3 IO1	Uses FLEXCOM3 IOSET 1 USART pins
14	UART3 IO2	Uses FLEXCOM3 IOSET 2 USART pins
15	UART3 IO3	Uses FLEXCOM3 IOSET 3 USART pins
16	UART3 IO4	Uses FLEXCOM3 IOSET 4 USART pins
17	UART3 IO5	Uses FLEXCOM3 IOSET 5 USART pins
18	UART4 IO1	Uses FLEXCOM4 IOSET 1 USART pins
19	UART4 IO2	Uses FLEXCOM4 IOSET 2 USART pins
20	UART4 IO3	Uses FLEXCOM4 IOSET 3 USART pins
21	UART4 IO4	Uses FLEXCOM4 IOSET 4 USART pins
22	UART4 IO5	Uses FLEXCOM4 IOSET 5 USART pins
23	UART5 IO1	Uses FLEXCOM5 IOSET 1 USART pins

Value	Name	Description
24	UART5 IO2	Uses FLEXCOM5 IOSET 2 USART pins
25	UART5 IO3	Uses FLEXCOM5 IOSET 3 USART pins
26	UART5 IO4	Uses FLEXCOM5 IOSET 4 USART pins
27	UART5 IO5	Uses FLEXCOM5 IOSET 5 USART pins
28	UART6 IO1	Uses FLEXCOM6 IOSET 1 USART pins
29	UART6 IO2	Uses FLEXCOM6 IOSET 2 USART pins
30	UART6 IO3	Uses FLEXCOM6 IOSET 3 USART pins
31	UART6 IO4	Uses FLEXCOM6 IOSET 4 USART pins
32	UART6 IO5	Uses FLEXCOM6 IOSET 5 USART pins
33	UART7 IO1	Uses FLEXCOM7 IOSET 1 USART pins
34	UART7 IO2	Uses FLEXCOM7 IOSET 2 USART pins
35	UART7 IO3	Uses FLEXCOM7 IOSET 3 USART pins
36	UART7 IO4	Uses FLEXCOM7 IOSET 4 USART pins
37	UART7 IO5	Uses FLEXCOM7 IOSET 5 USART pins
38	UART8 IO1	Uses FLEXCOM8 IOSET 1 USART pins
39	UART8 IO2	Uses FLEXCOM8 IOSET 2 USART pins
40	UART8 IO3	Uses FLEXCOM8 IOSET 3 USART pins
41	UART8 IO4	Uses FLEXCOM8 IOSET 4 USART pins
42	UART8 IO5	Uses FLEXCOM8 IOSET 5 USART pins
43	UART9 IO1	Uses FLEXCOM9 IOSET 1 USART pins
44	UART9 IO2	Uses FLEXCOM9 IOSET 2 USART pins
45	UART9 IO3	Uses FLEXCOM9 IOSET 3 USART pins
46	UART9 IO4	Uses FLEXCOM9 IOSET 4 USART pins
47	UART9 IO5	Uses FLEXCOM9 IOSET 5 USART pins
48	UART10 IO1	Uses FLEXCOM10 IOSET 1 USART pins
49	UART10 IO2	Uses FLEXCOM10 IOSET 2 USART pins
50	UART10 IO3	Uses FLEXCOM10 IOSET 3 USART pins
51	UART10 IO4	Uses FLEXCOM10 IOSET 4 USART pins
52	UART10 IO5	Uses FLEXCOM10 IOSET 5 USART pins
53	UART11 IO1	Uses FLEXCOM11 IOSET 1 USART pins
54	UART11 IO2	Uses FLEXCOM11 IOSET 2 USART pins
55	UART11 IO3	Uses FLEXCOM11 IOSET 3 USART pins
56	UART11 IO4	Uses FLEXCOM11 IOSET 4 USART pins
57	UART11 IO5	Uses FLEXCOM11 IOSET 5 USART pins

19.1.4.7 QSPI Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
1	QSPI interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, QSPI

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, QSPI

19.1.4.8 QSPI Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

⚠WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	QSPI_MODE
Access								
Reset								

Bits 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU
DO NOT USE

Value	Description
0	QSPI mode – QSPI memory connected to the QSPI controller
1	SPI mode enabled – SPI memory connected to the QSPI controller

Bit 0 – QSPI_MODE QSPI Mode

19.1.4.9 SDMMC Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

⚠WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
3	SDMMC interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, SDMMC
1	IP instance 1, SDMMC
2	IP instance 2, SDMMC

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, SDMMC

19.1.4.10 SDMMC Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ENABLE	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 15:8 – WPKEY[7:0] Write Protect Key

Value	Name	Description
`0x96`	PASSWD	If any other value is written in this field, all the other bit values are ignored.

Bit 7 – ENABLE Card Detect Enable

Value	Description
`0`	Card detect disable, the ROM code does not use any card detect pin and directly tries to boot from the memory connected to the SDMMC controller.
`1`	Card detect enable, the ROM code checks the level of the card detect pin. If the level is 0, the ROM code tries to boot from the memory connected to the SDMMC controller. If the level is 1, the ROM code skips the SDMMC controller and jumps to the next interface in the boot sequence.

Bits 0, 1, 2, 3, 4, 5, 6 – DNU DO NOT USE

19.1.4.11 NVM Boot Sequence

The ROM code performs the initialization and valid code detection for external memories as described below when the memory interface boot is enabled in the Boot Configuration packet.

Figure 19-3. NVM Boot Sequence

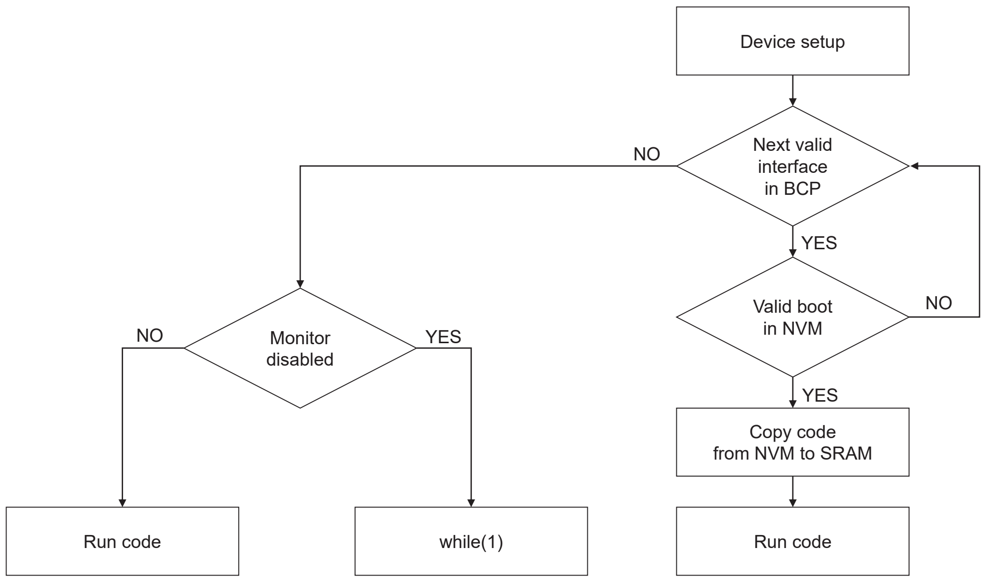
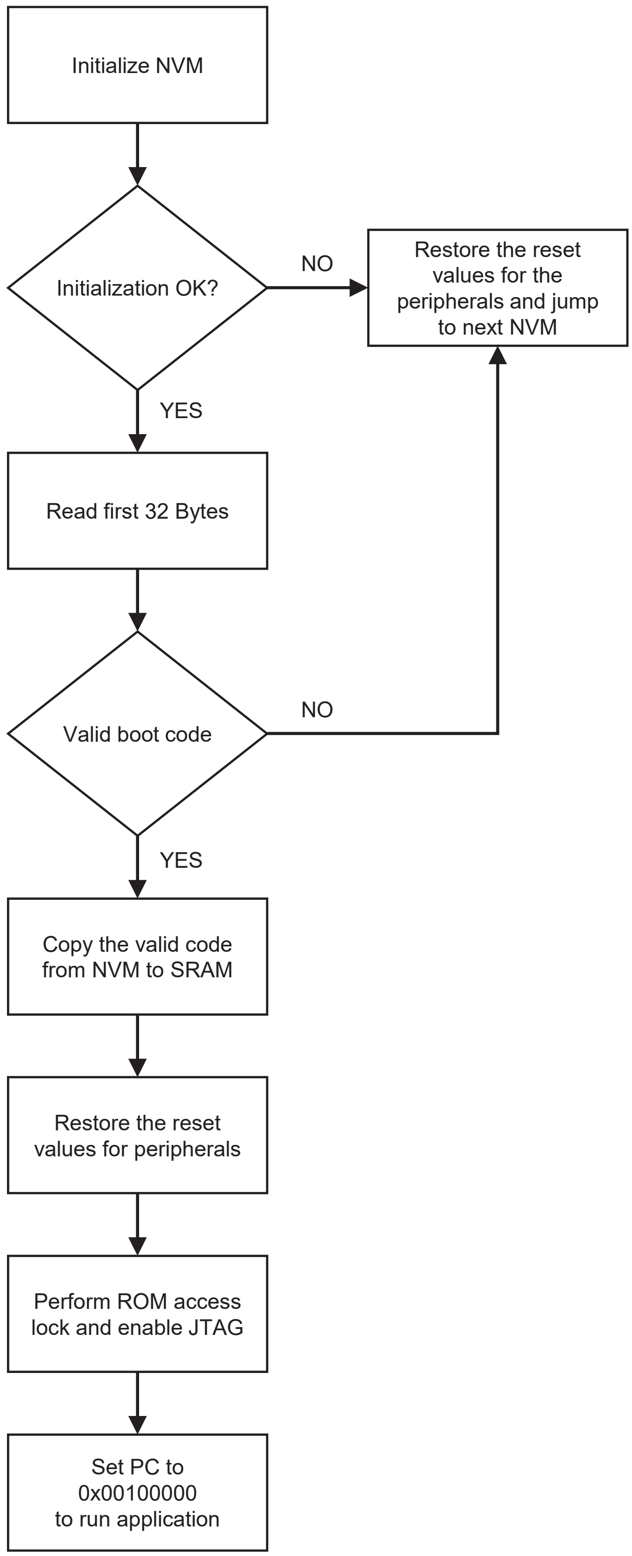


Figure 19-4. NVM Boot Diagram



19.1.4.12 Valid Code Detection

Two types of valid code detection are available:

- Arm Exception Vector Check
- boot.bin File Check

19.1.4.12.1 Arm Exception Vector Check

The ROM code analyzes the first 28 bytes corresponding to the first seven Arm exception vectors. Except for the sixth vector, these bytes must implement the Arm instructions for either branch or load PC with PC-relative addressing.

Figure 19-5. LDR Opcode

31			28	27			24	23			20	19 … 16	15 … 12	11 … 0
1	1	1	0	0	1	I	P	U	1	W	0	Rn	Rd	Offset

Figure 19-6. B Opcode

31			28	27			24	23 … 0
1	1	1	0	1	0	1	0	Offset (24 bits)

Unconditional instruction: 0xE for bits 31 to 28.

Load PC with the PC-relative addressing instruction:

- Rn = Rd = PC = 0xF
- I==0 (12-bit immediate value)
- P==1 (pre-indexed)
- U offset added (U==1) or subtracted (U==0)
- W==1

The sixth vector, at offset 0x14, contains the size of the image to download. The user must replace this vector with the user's own vector. This procedure is described below.

Figure 19-7. Arm Vector 6 Structure

31 … 0
Size of the code to download in bytes

The value must be lower than 64 Kbytes

The following is an example of valid vectors:

```
00          ea000006                B 0x20
04          eafffffe                B 0x04
08          eafffffe                B 0x08
0c          eafffffe                B 0x0c
10          eafffffe                B 0x10
14          00001234                ← Code size = 4660 bytes
18          eafffffe                B 0x18
```

19.1.4.12.2 boot.bin File Check

This method is the one used on FAT-formatted SD Card and e.MMC. The boot program must be a file named "boot.bin" written in the file system root directory. Its size must not exceed the maximum size allowed: 64 Kbytes (0x10000). The Arm exception vectors at offset 0 in the "boot.bin" file must also pass the test described in Arm Exception Vector Check.

19.1.4.13 Detailed Memory Boot Procedures

19.1.4.13.1 SD Card/e.MMC Boot

In the case of activated Card Detect pin in the Boot Configuration Packet, and if the level of the Card Detect pin is high, no communication with SD Card/e.MMC is performed (no IOs toggling). Otherwise, the SD Card/e.MMC access is initiated (IOs toggling).

Supported SD Card Devices

SD Card Boot supports all SD Card memories compliant with the SD Memory Card Specification V3.0. This includes SDMMC cards.

e.MMC with Boot Partition

The ROM code first checks if the e.MMC boot partition is enabled. If enabled, the ROM code reads the first 64 Kbytes of the boot partition and copy them into the internal SRAM.

FAT Filesystem Boot

If no boot partition is enabled on an e.MMC, the boot process continues with a standard SD Card/e.MMC detection, and the ROM code looks for a “boot.bin” file in the root directory of a FAT12/16/32 file system.

19.1.4.13.2 SPI Flash Boot

The ROM code allows booting from SPI memories connected to the QSPI controller. The QSPI_MODE bit must be set in the Boot Configuration Packet.

SPI Flash Boot supports only SPI Serial Flash devices.

The SPI Flash read is done by means of a Continuous Read command from the address 0x0. This command is 0x0B for Serial Flash devices.

Supported Serial Flash Devices

The SPI Flash Boot program supports all SPI Serial Flash devices responding correctly to the Get Status and Continuous Read commands.

19.1.4.13.3 QSPI NOR Flash Boot

Hardware Considerations

The ROM code configures the hardware so that:

- the QSPI controller uses SPI Mode 0 (CPOL = 0 and CPHA = 0),
- the QSPIx_SCK clock frequency is ≤ 50 MHz,
- QSPIx_SCK and QSPIx_CS do not use any internal pull-up/pull-down resistor,
- each QSPIx_IO{0,1,2,3} uses the PIO controller’s internal pull-up resistor.

Software Considerations

Before reading any data, the ROM code sends a software reset to the QSPI NOR memory. Then the ROM code looks for the Serial Flash Discoverable Parameters (SFDP) of the QSPI NOR memory, if available, to learn the parameters (instruction op code, timing settings) required to read the user-programmed boot file.

If SFDP tables are not available, the ROM code uses hard-coded values as fallback settings to read the boot file.

The ROM code supports any QSPI NOR memory which can provide its SFDP as defined in the JEDEC JESD216B standard.

The supported revisions of this JEDEC standard are:

- JESD216 (version 1.0)
- JESD216 rev. A (version 1.5)
- JESD216 rev. B (version 1.6)

Refer to the QSPI NOR memory data sheet to check compliance with the above JEDEC JESD216 standard revisions/versions.

QSPI NOR Memories with SFDP (JEDEC JESD216x Compliant)

The ROM code reads the memory SFDP tables to learn the factory settings (instruction op code, number of dummy cycles, etc.). The ROM code also reads bits[22:20] in DWORD15 from the Basic Flash Parameter table (refer to the JEDEC JESD216B specification) to select and then execute the relevant procedure, if any, to set the Quad Enable (QE) bit in some internal register of the QSPI NOR memory.

For most memory manufacturers, the QE bit is non-volatile and must be set before performing any Quad SPI command. This is the only persistent setting that the ROM code may change in the internal registers of the QSPI NOR memory. All other settings are kept unchanged.

Refer to the QSPI NOR memory data sheet to find which value was chosen by the memory manufacturer and written into the SFDP tables.

Finally, the ROM code reads the boot file from the data area of the QSPI NOR memory, and then continues its boot procedure.

QSPI NOR Memories without SFDP

This section only applies when the ROM code fails to read the SFDP tables from the QSPI NOR memory.

The ROM code reads the JEDEC ID of the QSPI NOR memory, and then selects the read settings based on the manufacturer ID (first byte of the JEDEC ID) from the following hard-coded values:

Table 19-2. QSPI NOR Memories Settings

	Cypress (01h)	Micron (20h)	Macronix (C2h)	Winbond (EFh)	Others
Fast Read protocol	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-1-1
Fast Read op code	EBh	EBh	EBh	EBh	0Bh
Address width	24 bits	24 bits	24 bits	24 bits	24 bits
Number of mode clock cycles	2	1	2	2	0
Number of wait states	4	9	4	4	8
Value of mode cycles to exit the 0-4-4 mode (normal read)	00h	1h	00h	FFh	N/A

Those hard-coded parameters give a last chance to the ROM code to boot from a QSPI NOR memory in either normal mode.

Table 19-3. QSPI NOR Memories Supported by ROM Code (Non Exhaustive)

Manufacturer	Memories
Microchip (SST)	SST26VF016B SST26VF032B SST26VF032BA SST26VF064B
Micron	N25Q128A N25Q128A13ESF N25Q256A13ESF N25Q512A13 MT25QL01G

...........continued

Manufacturer	Memories
Macronix	MX25V4035FM2I MX25V8035FM2I MX25V1635FM2I MX25L3233FM2I-08G MX25L3273FM2I-08G MX25L6433FM2I-08G MX25L6473FM2I-08G MX25L12835FM2I-10G MX25L12845GMI-08G MX25L12873GM2I-08G MX25L25645G MX25L25673G MX25L51245GMI-10G MX66L1G45GMI-08G
Spansion	S25FL127 S25FL164 S25FL512
Winbond	W25M512

Note: For an updated list of memories, refer to the application note “Booting from External Non-Volatile Memory (NVM) on SAMA7G5 MPUs” (document no. DS00004408) available on www.microchip.com.

19.1.4.14 Hardware and Software Constraints

The following table provides clock frequencies configured by the ROM code during boot.

Table 19-4. Clock Frequencies During External Memory Boot Sequence

Clock	Frequency
SYSPLL	378 MHz
CPU_CLK / CPUPLL	570 MHz
MCK0	142 MHz
MCK1	189 MHz
MCK4	378 MHz
SDMMC (init/operational)	400 kHz / 25 MHz
SPI	47 MHz
QSPI	47 MHz

The NVM drivers use several PIOs in Peripheral mode to communicate with external memory devices. Care must be taken when these PIOs are used by the application. The connected devices could be unintentionally driven at boot time, and thus electrical conflicts between the output pins used by the NVM drivers and the connected devices could occur.

The following table contains a list of pins that are driven during the boot program execution. These pins are driven during the boot sequence for a period of less than 1 second if no correct boot program is found. The drive strength of pull-up I/O pins is set to low while the pins are used in Peripheral mode by the ROM code.

Before performing the jump to the application in the internal SRAM, all the PIOs and peripherals used in the boot program are set to their reset state.

Table 19-5. PIO Driven During Boot Program Execution

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
SD Card/e.MMC	SDMMC_0	1	SDMMC0_CK	PIO_PA0A	–
			SDMMC0_CMD	PIO_PA1A	X
			SDMMC0_DAT0	PIO_PA3A	X
			SDMMC0_DAT1	PIO_PA4A	X
			SDMMC0_DAT2	PIO_PA5A	X
			SDMMC0_DAT3	PIO_PA6A	X
	SDMMC_1	1	SDMMC1_CMD	PIO_PB29A	X
			SDMMC1_CK	PIO_PB30A	–
			SDMMC1_DAT0	PIO_PB31A	X
			SDMMC1_DAT1	PIO_PC0A	X
			SDMMC1_DAT2	PIO_PC1A	X
			SDMMC1_DAT3	PIO_PC2A	X
	SDMMC_2	1	SDMMC2_CMD	PIO_PD3A	X
			SDMMC2_CK	PIO_PD4A	–
			SDMMC2_DAT0	PIO_PD5A	X
			SDMMC2_DAT1	PIO_PD6A	X
			SDMMC2_DAT2	PIO_PD7A	X
			SDMMC2_DAT3	PIO_PD8A	X
QSPI Flash	QSPI_0	1	QIO3	PIO_PB9A	X
			QIO2	PIO_PB10A	X
			QIO1	PIO_PB11A	X
			QIO0	PIO_PB12A	X
			QCS	PIO_PB13A	–
			QSCK	PIO_PB14A	–
			QSCKN	PIO_PB15A	–
			QIO4	PIO_PB16A	X
			QIO5	PIO_PB17A	X
			QIO6	PIO_PB18A	X
			QIO7	PIO_PB19A	X
	QSPI_1	1	QIO3	PIO_PB22A	X
			QIO2	PIO_PB23A	X
			QIO1	PIO_PB24A	X
			QIO0	PIO_PB25A	X
			QCS	PIO_PB26A	–
			QSCK	PIO_PB27A	–

...........continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
Console and Monitor	FLEXCOM0_UART	1	DTXD	PIO_PA0B	–
			DRXD	PIO_PA1A	X
	FLEXCOM0_UART	2	DTXD	PIO_PD3B	–
			DRXD	PIO_PD4B	X
	FLEXCOM0_UART	3	DTXD	PIO_PD10F	–
			DRXD	PIO_PD11F	X
	FLEXCOM0_UART	4	DTXD	PIO_PE3B	–
			DRXD	PIO_PE4B	X
	FLEXCOM1_UART	1	DTXD	PIO_PA5B	–
			DRXD	PIO_PA6B	X
	FLEXCOM1_UART	2	DTXD	PIO_PD12B	–
			DRXD	PIO_PD13B	X
	FLEXCOM1_UART	3	DTXD	PIO_PA2F	–
			DRXD	PIO_PA3F	X
	FLEXCOM1_UART	4	DTXD	PIO_PC9F	–
			DRXD	PIO_PC10F	X
	FLEXCOM2_UART	1	DTXD	PIO_PA7B	–
			DRXD	PIO_PA8B	X
	FLEXCOM2_UART	2	DTXD	PIO_PD17B	–
			DRXD	PIO_PD18B	X
	FLEXCOM2_UART	3	DTXD	PIO_PA4F	–
			DRXD	PIO_PA5F	X
	FLEXCOM2_UART	4	DTXD	PIO_PC11F	–
			DRXD	PIO_PC12F	X
	FLEXCOM2_UART	5	DTXD	PIO_PD14F	–
			DRXD	PIO_PD15F	X
	FLEXCOM3_UART	1	DTXD	PIO_PA15B	–
			DRXD	PIO_PA16B	X
	FLEXCOM3_UART	2	DTXD	PIO_PC0B	–
			DRXD	PIO_PC1B	X
	FLEXCOM3_UART	3	DTXD	PIO_PA6F	–
			DRXD	PIO_PA7F	X
	FLEXCOM3_UART	4	DTXD	PIO_PC13F	–
			DRXD	PIO_PC14F	X
	FLEXCOM3_UART	5	DTXD	PIO_PD16F	–
			DRXD	PIO_PD17F	X
	FLEXCOM4_UART	1	DTXD	PIO_PA20B	–
			DRXD	PIO_PA21B	X
	FLEXCOM4_UART	2	DTXD	PIO_PC2B	–
			DRXD	PIO_PC3B	X
	FLEXCOM4_UART	3	DTXD	PIO_PA8F	–
			DRXD	PIO_PA9F	X
	FLEXCOM4_UART	4	DTXD	PIO_PC15F	–
			DRXD	PIO_PC16F	X
	FLEXCOM4_UART	5	DTXD	PIO_PD18F	–
			DRXD	PIO_PD19F	X

...........continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
Console and Monitor	FLEXCOM5_UART	1	DTXD	PIO_PA28B	–
			DRXD	PIO_PA29B	X
	FLEXCOM5_UART	2	DTXD	PIO_PD25B	–
			DRXD	PIO_PD26B	X
	FLEXCOM5_UART	3	DTXD	PIO_PA10F	–
			DRXD	PIO_PA11F	X
	FLEXCOM5_UART	4	DTXD	PIO_PC17F	–
			DRXD	PIO_PC18F	X
	FLEXCOM5_UART	5	DTXD	PIO_PD20F	–
			DRXD	PIO_PD21F	X
	FLEXCOM6_UART	1	DTXD	PIO_PB2B	–
			DRXD	PIO_PB1B	X
	FLEXCOM6_UART	2	DTXD	PIO_PD30B	–
			DRXD	PIO_PD31B	X
	FLEXCOM6_UART	3	DTXD	PIO_PA12F	–
			DRXD	PIO_PA13F	X
	FLEXCOM6_UART	4	DTXD	PIO_PC19F	–
			DRXD	PIO_PC20F	X
	FLEXCOM6_UART	5	DTXD	PIO_PC21F	–
			DRXD	PIO_PC22F	X
	FLEXCOM7_UART	1	DTXD	PIO_PB23B	–
			DRXD	PIO_PB24B	X
	FLEXCOM7_UART	2	DTXD	PIO_PC7B	–
			DRXD	PIO_PC8B	X
	FLEXCOM7_UART	3	DTXD	PIO_PA25F	–
			DRXD	PIO_PA26F	X
	FLEXCOM7_UART	4	DTXD	PIO_PB28F	–
			DRXD	PIO_PB29F	X
	FLEXCOM7_UART	5	DTXD	PIO_PC23F	–
			DRXD	PIO_PC24F	X
	FLEXCOM8_UART	1	DTXD	PIO_PB8B	–
			DRXD	PIO_PB9B	X
	FLEXCOM8_UART	2	DTXD	PIO_PC14B	–
			DRXD	PIO_PC13B	X
	FLEXCOM8_UART	3	DTXD	PIO_PA27F	–
			DRXD	PIO_PA28F	X
	FLEXCOM8_UART	4	DTXD	PIO_PB30F	–
			DRXD	PIO_PB31F	X
	FLEXCOM8_UART	5	DTXD	PIO_PD2F	–
			DRXD	PIO_PD3F	X
	FLEXCOM9_UART	1	DTXD	PIO_PB13B	–
			DRXD	PIO_PB14B	X
	FLEXCOM9_UART	2	DTXD	PIO_PC18B	–
			DRXD	PIO_PC19B	X
	FLEXCOM9_UART	3	DTXD	PIO_PA29F	–
			DRXD	PIO_PA30F	X

...........continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
	FLEXCOM9_UART	4	DTXD	PIO_PC0F	–
			DRXD	PIO_PC1F	X
	FLEXCOM9_UART	5	DTXD	PIO_PD4F	–
			DRXD	PIO_PD5F	X
	FLEXCOM10_UART	1	DTXD	PIO_PB18B	–
			DRXD	PIO_PB19B	X
	FLEXCOM10_UART	2	DTXD	PIO_PC30B	–
			DRXD	PIO_PC31B	X
	FLEXCOM10_UART	3	DTXD	PIO_PA31F	–
			DRXD	PIO_PB0F	X
	FLEXCOM10_UART	4	DTXD	PIO_PC2F	–
			DRXD	PIO_PC3F	X
	FLEXCOM10_UART	5	DTXD	PIO_PC6F	–
			DRXD	PIO_PC7F	X
	FLEXCOM11_UART	1	DTXD	PIO_PB3B	–
			DRXD	PIO_PB4B	X
	FLEXCOM11_UART	2	DTXD	PIO_PD0B	–
			DRXD	PIO_PD1B	X
	FLEXCOM11_UART	3	DTXD	PIO_PB1F	–
			DRXD	PIO_PB2F	X
	FLEXCOM11_UART	4	DTXD	PIO_PC4F	–
			DRXD	PIO_PC5F	X
	FLEXCOM11_UART	5	DTXD	PIO_PD8F	–
			DRXD	PIO_PD9F	X

19.1.5 Standard Monitor

This part of the ROM code is executed when no valid code is found in any external Flash memory during the NVM boot sequence, and if the MON_DIS field is not set to 1 in the Boot Configuration Packet.

The main RC oscillator is still used as the main clock.

To use the USB device interface, an external quartz or accurate clock must be connected to the chip.

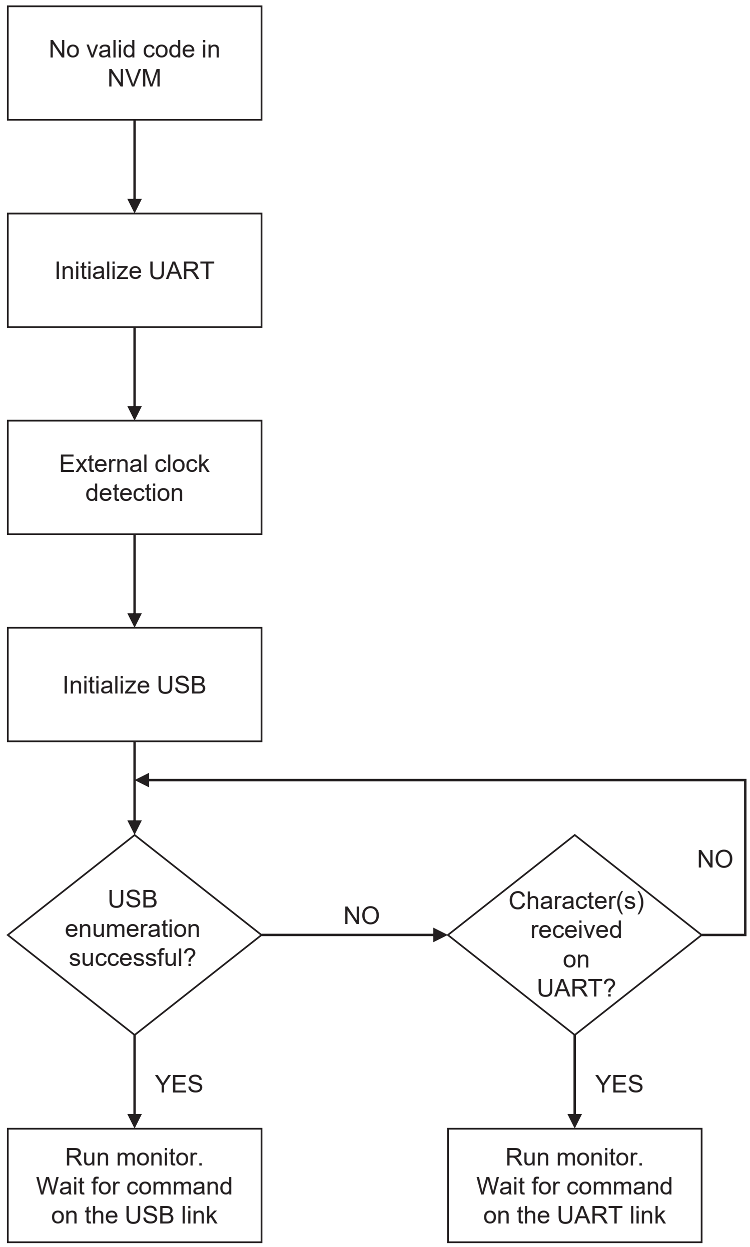
In case no external clock source is supplied, the USB interface is not activated and only a UART connection can be used as a monitor.

The monitor steps are:

- Initialize UART and USB.
- Check if USB device enumeration occurred.
- Check if characters are received on the UART.

Once the communication interface is identified, the monitor runs in an infinite loop waiting for different commands as listed in Command List.

Figure 19-8. Standard Monitor



19.1.5.1 Command List

Table 19-6. Commands Available Through Standard Monitor

Command	Action	Argument(s)	Example
N	Set Normal Mode	No argument	**N**#
T	Set Terminal Mode	No argument	**T**#
O	Write a byte	Address, Value#	**O**200001,CA#
o	Read a byte	Address,#	**o**200001,#

...........continued			
Command	**Action**	**Argument(s)**	**Example**
H	Write a halfword	Address, Value#	**H**200002,CAFE#
h	Read a halfword	Address,#	**h**200002,#
W	Write a word	Address, Value#	**W**200000,CAFEDECA#
w	Read a word	Address,#	**w**200000,#
S	Send a file	Address,#	**S**200000,#
R	Receive a file	Address, NbOfBytes#	**R**200000,1234#
G	Go	Address#	**G**200200#
V	Display version	No argument	**V**#

- Mode commands:
  - Normal mode configures monitor to send/receive data in binary format.
  - Terminal mode configures monitor to send/receive data in ASCII format.
- Write commands: Writes a byte (**O**), a halfword (**H**) or a word (**W**) to the target.
  - Address: address in hexadecimal
  - Value: byte, halfword or word to write in hexadecimal
  - Output: ‘>’
- Read commands: Reads a byte (**o**), a halfword (**h**) or a word (**w**) from the target.
  - Address: address in hexadecimal
  - Output: the byte, halfword or word read in hexadecimal followed by ‘>’
- Send a file (**S**): Sends a file to a specified address.
  - Address: address in hexadecimal
  - Output: ‘>’

    Note: There is a timeout on this command which is reached when the prompt ‘>’ appears before the end of the command execution.
- Receive a file (**R**): Receives data into a file from a specified address.
  - Address: address in hexadecimal
  - NbOfBytes: number of bytes in hexadecimal to receive
  - Output: ‘>’
- Go (**G**): Jumps to a specified address and executes the code.
  - Address: address to jump to in hexadecimal
  - Output: ‘>’ once returned from the program execution. If the executed program does not handle the link register at its entry and does not return, the prompt is not displayed.
- Get Version (**V**): Returns the Boot Program version.
  - Output: version, date and time of ROM code followed by ‘>’

19.1.5.2 DBGU/UART Console Port

Communication is performed through the DBGU/UART port initialized to 115,200 bauds: 8 bits of data, no parity, 1-stop bit.

19.1.5.2.1 Xmodem Protocol

The Send and Receive File commands use the Xmodem protocol to communicate. Any terminal using this protocol can be used to send the application file to the target. The size of the binary file to send must be lower than half the SRAM size because the Xmodem protocol requires some SRAM memory in order to work.

The Xmodem protocol supported is the 128-byte length block. This protocol uses a two-character CRC16 to ensure detection of maximum bit errors.

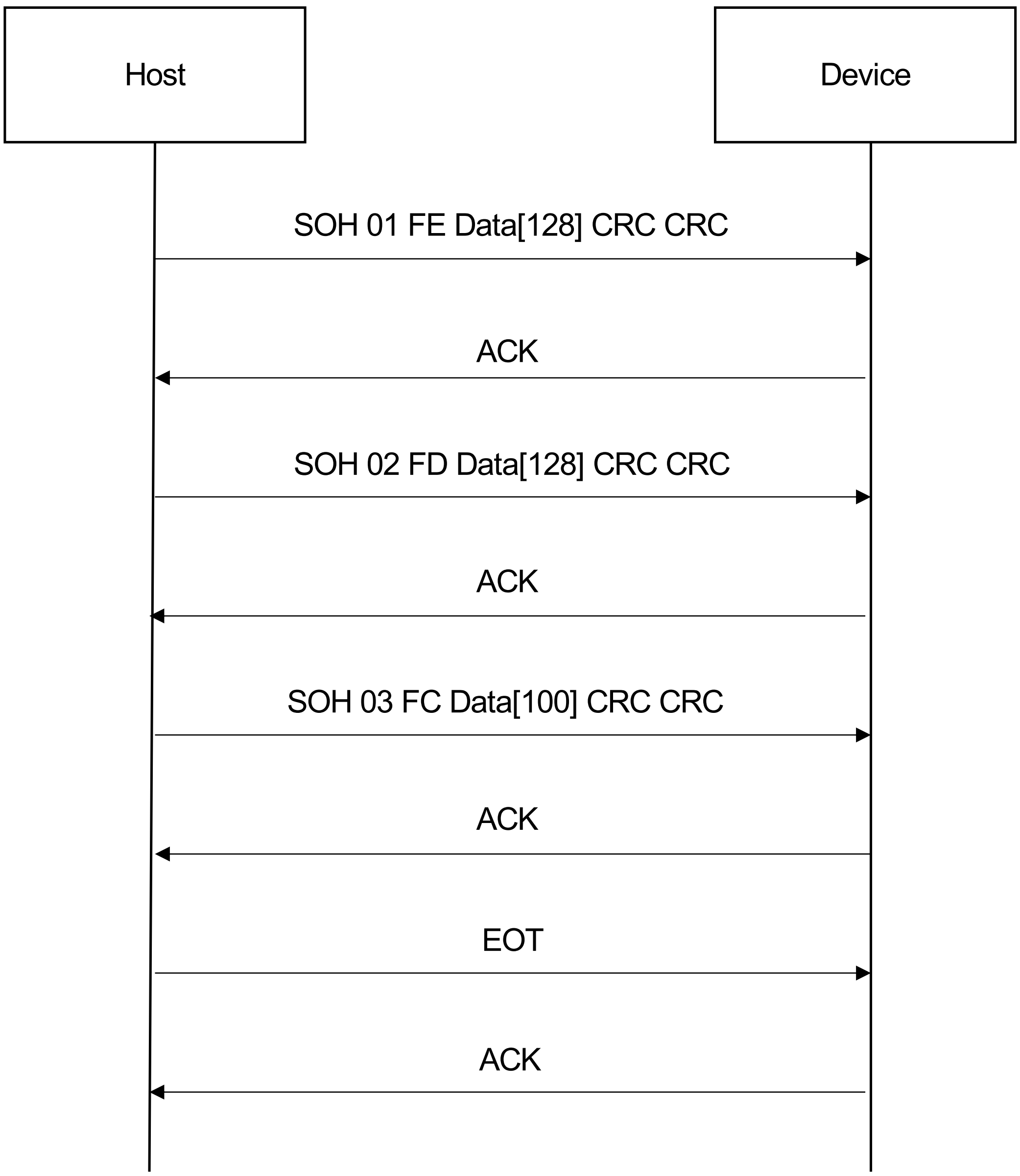
Xmodem protocol with CRC is supported by successful transmission reports provided both by a sender and by a receiver. Each transfer block is as follows:

<SOH><blk #><255-blk #><--128 data bytes--><checksum> in which:

- <SOH> = 01 hex
- <blk #> = binary number, starts at 01, increments by 1, and wraps 0FFH to 00H (not to 01)
- <255-blk #> = 1's complement of the blk#.
- <checksum> = 2-byte CRC16

The following figure shows a transmission using this protocol.

Figure 19-9. Xmodem Transfer Example



19.1.5.3 USB Device Port

19.1.5.3.1 Supported External Crystal/External Clocks

The SAMA7G5 chip supports an external crystal or external clock frequency at 16 MHz, 20 MHz, 24 MHz or 32 MHz to allow USB communication.

19.1.5.3.2 USB Class

The device uses the USB Communication Device Class (CDC) drivers to take advantage of the installed PC Serial Communication software to talk over the USB. The CDC is implemented in all releases of Microsoft Windows®, starting from Windows 98SE®. The CDC document, available at www.usb.org, describes how to implement devices such as ISDN modems and virtual COM ports.

The vendor ID is 0x03EB. The product ID is 0x6124. These references are used by the host operating system to mount the correct driver. On Windows systems, INF files contain the correspondence between vendor ID and product ID.

19.1.5.3.3 Enumeration Process

The USB protocol is a host/client protocol. The host starts the enumeration, sending requests to the device through the control endpoint. The device handles standard requests as defined in the USB specification.

Table 19-7. Handled Standard Requests

Request	Definition
GET_DESCRIPTOR	Returns the current device configuration value
SET_ADDRESS	Sets the device address for all future device accesses
SET_CONFIGURATION	Sets the device configuration
GET_CONFIGURATION	Returns the current device configuration value
GET_STATUS	Returns status for the specified recipient
SET_FEATURE	Used to set or enable a specific feature
CLEAR_FEATURE	Used to clear or disable a specific feature

The device also handles some class requests defined in the CDC class.

Table 19-8. Handled Class Requests

Request	Definition
SET_LINE_CODING	Configures DTE rate, stop bits, parity and number of character bits
GET_LINE_CODING	Requests current DTE rate, stop bits, parity and number of character bits
SET_CONTROL_LINE_STATE	RS-232 signal used to indicate to the DCE device that the DTE device is now present

Unhandled requests are stalled.

19.1.5.3.4 Communication Endpoints

Endpoint 0 is used for the enumeration process.

Endpoint 1 (64-byte Bulk OUT) and endpoint 2 (64-byte Bulk IN) are used as communication endpoints.

Monitor commands are sent by the host through Endpoint 1. If required, the message is split into several data payloads by the host driver.

If the command requires a response, the host sends IN transactions to pick up the response.

19.2 Secure Boot Strategies

19.2.1 Overview

The secure boot mode authenticates and deciphers a boot file stored in an external Non-Volatile Memory (NVM) prior to its execution. The boot file can be a bootstrap code or the user application. The secure boot ensures that only authorized code is executed, thus protecting the customer IP and providing a Root of Trust (RoT) in the hardware.

When the Secure Boot mode is enabled, the chip only allows booting on an authenticated and ciphered boot file. The boot file can be authenticated and deciphered in two ways:

- AES-CBC-CMAC mode:
  – Authentication is performed using the customer private CMAC key stored in the OTP memory, using the AES-CMAC algorithm.
  – Boot file decryption is performed using the customer private CBC key stored in the OTP memory, using the AES-CBC algorithm.
- AES-CBC-RSA mode:
  – Authentication is performed using the customer public key contained in the last X.509 certificate chain stored after the boot file.
  – Boot file decryption is performed using the customer private key stored in the OTP memory, using the AES-CBC algorithm.

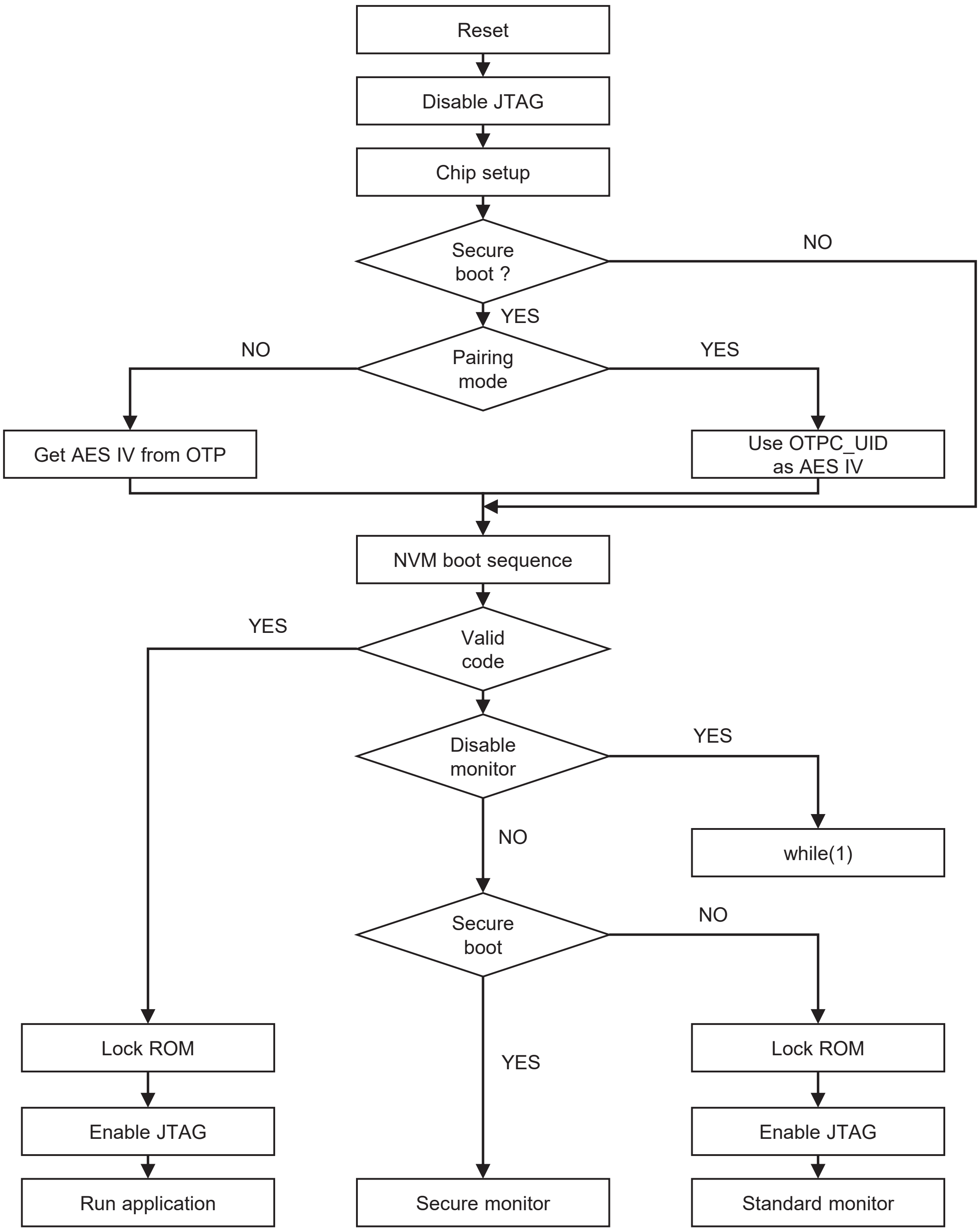
Additionally, a 'pairing' mode can be enabled. In this case, the customer private CBC key is diversified with the Chip Unique ID to decipher the boot file.

Encryption is supported by an AES (using HW acceleration); in addition, either symmetric (AES-CMAC) or asymmetric (RSA-based using X509 certificate chain) authentication is supported.

19.2.2 Flow Diagram

The standard and secure boot sequence flow is detailed in the following figure.

Figure 19-10. Boot Sequence Flow



19.2.3 Secure Boot Configuration

In addition to the Boot Configuration Packet content, the boot sequence flow is also controlled with the data stored in the Secure Boot Configuration packet.

The Secure Boot Configuration is stored in the Secure Boot Configuration Packet in the OTP user area. It defines all the information required to boot in Secure mode:

- Activate Secure mode
- Authentication mode
- Pairing mode option
- Number of keys
- etc.

See Secure Boot Configuration User Interface for a detailed description of the fields in the Secure Boot Configuration Packet.

During the prototyping phase, the content of the Boot Configuration and Secure Boot Configuration Packets can be overridden using the OTPC Emulation mode. See Boot Configuration Packet for more information.

19.2.4 Secure Boot Configuration User Interface

Table 19-9. Secure Boot Configuration Packet

Offset	Name	Description
0x00	SEC_MODE_EN	Secure mode enabled
0x04	AUTH_MODE	Authentication mode selection
0x08	PAIRING_MODE_EN	Pairing mode enabled
0x0C	KEY_WRITTEN	Key written
0x10	IV	AES Initialization Vector address
0x14	RSA_HASH	RSA Root Certificate public key hash address
0x18	CBC_KEY	AES CBC key
0x1C	CMAC_KEY	AES CMAC key

19.2.4.1 Secure Boot Mode Enable

Name: SEC_BOOT_MODE_EN

Bit	31	30	29	28	27	26	25	24
	SECURE_BOOT_MODE_ENABLE							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	SECURE_BOOT_MODE_ENABLE							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	SECURE_BOOT_MODE_ENABLE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SECURE_BOOT_MODE_ENABLE							

SECURE_BOOT_MODE_ENABLE

0: Secure Boot mode is disabled.

1: Secure Boot mode is enabled.

19.2.4.2 Authentication Mode

Name: AUTH_MODE

Bit	31	30	29	28	27	26	25	24
	AUTHENTICATION_MODE							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	AUTHENTICATION_MODE							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	AUTHENTICATION_MODE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	AUTHENTICATION_MODE							

AUTHENTICATION_MODE

0: AES-CMAC is used for authentication.

1: RSA signature verification is used for authentication.

19.2.4.3 Pairing Mode Enable

Name: PAIRING_MODE_EN

Bit	31	30	29	28	27	26	25	24
	PAIRING_MODE_ENABLE							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PAIRING_MODE_ENABLE							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PAIRING_MODE_ENABLE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PAIRING_MODE_ENABLE							

PAIRING_MODE_ENABLE

0: Pairing mode is disabled.

Different from 0: Pairing mode is enabled.

19.2.4.4 Key Written

Name: KEY_WRITTEN

Bit	31	30	29	28	27	26	25	24
	KEY_WRITTEN							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	KEY_WRITTEN							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	KEY_WRITTEN							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	KEY_WRITTEN							

KEY_WRITTEN

0: Customer secret keys are not written in the OTP memory.

Different from 0: Customer secret keys are written in the OTP memory.

19.2.4.5 Initialization Vector Address

Name: IV

Bit	31	30	29	28	27	26	25	24
	IV_ADDRESS							
Bit	23	22	21	20	19	18	17	16
	IV_ADDRESS							
Bit	15	14	13	12	11	10	9	8
	IV_ADDRESS							
Bit	7	6	5	4	3	2	1	0
	IV_ADDRESS							

IV_ADDRESS: Address of the OTP packet containing the AES Initialization Vector

19.2.4.6 RSA Public Key Hash Address

Name: RSA_HASH

Bit	31	30	29	28	27	26	25	24
	RSA_HASH_ADDRESS							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	RSA_HASH_ADDRESS							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	RSA_HASH_ADDRESS							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RSA_HASH_ADDRESS							

RSA_HASH_ADDRESS: Address of the OTP packet containing the hash of the root certificate CA

19.2.4.7 AES-CBC Key Address

Name: CBC_KEY

Bit	31	30	29	28	27	26	25	24
	AES_CBC_KEY_ADDRESS							
Bit	23	22	21	20	19	18	17	16
	AES_CBC_KEY_ADDRESS							
Bit	15	14	13	12	11	10	9	8
	AES_CBC_KEY_ADDRESS							
Bit	7	6	5	4	3	2	1	0
	AES_CBC_KEY_ADDRESS							

AES_CBC_KEY_ADDRESS: Address of the OTP packet containing the secret key for AES-CBC

19.2.4.8 AES-CMAC Key Address Register

Name: CMAC_KEY

Bit	31	30	29	28	27	26	25	24
	AES_CMAC_KEY_ADDRESS							
Bit	23	22	21	20	19	18	17	16
	AES_CMAC_KEY_ADDRESS							
Bit	15	14	13	12	11	10	9	8
	AES_CMAC_KEY_ADDRESS							
Bit	7	6	5	4	3	2	1	0
	AES_CMAC_KEY_ADDRESS							

AES_CMAC_KEY_ADDRESS: Address of the OTP packet containing the secret key for AES-CMAC

19.2.5 Secure Valid Code Detection

The valid code detection in Secure mode is similar to the one in Standard Boot mode. However, additional checks and operations are done.

If the initialization of NVM is successful, the ROM code reads and deciphers the first 32 bytes of the potential secure boot file to find a valid Arm exception vector table (see Arm Exception Vector Check). From the sixth vector, the ROM code extracts the size of the boot file including its signature. When the AUTH_MODE field is set to 1 in the Secure Boot Configuration Packet, the size of the X.509 certificate chain is also extracted. The boot file size must be 16 bytes aligned (AES block size). The sum of the boot file and certificate sizes must be lower than 64 Kbytes. If these first validations fail, the ROM code restores the memory interface PIO and settings to their reset values and then tries to boot on the next NVM.

Otherwise, the total size is used to copy the boot file, its signature and the X.509 certificate chain from the NVM into the internal SRAM.

Next, depending on the AUTH_MODE field in the Secure Boot Configuration Packet, either the AES-CMAC digest or the RSA signature of the boot file is checked. As usual, if this final validation fails, the peripheral is reset and the ROM code jumps to the next NVM.

If the boot file verification passes, the ROM code deciphers the boot file stored in the internal SRAM.

Since the NVM interface is no longer needed, the ROM code restores the interface registers and IO reset values.

Finally, the ROM code locks access to ROM area, enables JTAG and then branches to 0x00100000 to execute the boot file.

19.2.5.1 6th Vector Format

When the AUTH_MODE field is cleared in the Secure Boot Configuration Packet, the sixth entry of the Arm exception vector table stores the size of the boot file in bytes, including the 16 bytes of its AES-CMAC signature.

When the AUTH_MODE field is set to 1 in the Boot Configuration word, the sixth entry of the Arm exception vector table stores both the size of the boot file including its RSA signature and the size of the X.509 certificate chain. The size of the RSA signature is also encoded.

Bit	31	30	29	28	27	26	25	24
	CERTIFICATE_SIZE							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CERTIFICATE_SIZE					SIG_CODE		BT_SIZE
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	BT_SIZE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	BT_SIZE							

BT_SIZE: Boot file size, in bytes

SIG_CODE: Code for the RSA signature length

00: 2048 bits

01: 3072 bits

10: 4096 bits

11: RFU

CERTIFICATE_SIZE: Size of the of X.509 certificate chain, in bytes

19.2.5.2 e.MMC/SD Card File System

In Secure Boot mode, the ROM code looks for a `boot.cip` file in the root directory of a FAT12/16/32 file system. The secure boot file, its signature and the chain of X.509 certificates are stored in the `boot.cip` file.

19.2.6 Encryption, Decryption and Authentication

For security reasons, the customer's two 256-bit secret keys are transferred only through the Host Key Bus from the OTPC key packet to the AES hardware. The customer's AES Initialization Vector read from the OTP User area and the first key are used by the AES-CBC algorithm, whereas the second is dedicated to the AES-CMAC computation.

Encryption and decryption are processed using the NIST recommended AES CBC mode defined in NIST Special Publication 800-38A [NIST_MODE_OP].

When selected, the Message Authentication Code is processed using the NIST recommended CMAC. The CMAC used as a Message Authentication Code (MAC) is the CMAC based on the AES. The AES-CMAC outputs a 128-bit digest.

Otherwise, the authentication is secured by an RSA signature. The public key cryptographic implementation then relies on X.509 certificates. These certificates are chained and stored right after the boot file and its signature in the NVM.

The first certificate in the chain (at the lowest address in memory) is called the "root certificate". The modulus and the exponent are extracted from the public key stored in the root certificate. A SHA-256 digest is computed on the concatenation of the modulus and the exponent. This 256-bit digest is compared to the "RSA Hash" to validate the root certificate.

Then, except for the root certificate, every certificate is signed by the private key associated with the previous certificate in the chain. So, its previous certificate is used to validate a certificate in a recursive process.

Finally, the boot file is signed with the private key associated with the last certificate. So, the last certificate is used to validate the boot file signature.

19.2.7 Secure Boot Mode Configuration

By default, the device operates a Standard Boot sequence (see Standard Boot Strategies). In this mode, the chip can boot on a standard binary program present in any external non-volatile memory.

To configure the device to operate a Secure Boot sequence and to be able to boot a ciphered application, at least two steps must be followed, the third being done only when using the RSA signature:

1. Set the chip in Secure mode.
2. Send the customer key.
3. (If AES-RSA mode is selected) Send RSA Hash.

These steps are mandatory prior to programming a boot file in an external NVM and can only be done once.

To set the chip in Secure mode, a valid Secure Boot Configuration packet must be written.

19.2.8 Programming a Boot File

If the boot file is ciphered with an incorrect key or has an incorrect signature, the Secure Boot sequence will not detect this boot file as valid.

The boot file cipher process can be done by the Secure SAM-BA cipher tool.

19.2.9 Secure Monitor

When no bootable program is found in any external NVM memory during the boot sequence and Secure mode is enabled, the ROM code executes the secure monitor.

The communication link is a USB connection in CDC mode (seen as a serial COM port on the OS).

The set of commands implemented in the secure monitor enables the user to send all the commands to program the customer key and the HASH of the public RSA key and to configure the Secure Boot mode.

19.2.9.1 Generic Commands

The command format is similar to the standard monitor command format, and is composed of command opcode and arguments, and ends with a '#' character. Depending on the command, a data payload can be added after the command. In this case, the payload must be sent after the secure monitor has acknowledged the command and instructed how the data payload must be split (size of the payload).

19.2.9.2 Command Format

The commands are formatted as follows:

____,________,________,________,__#

`opcode addrxx length not_used rw`

where:

- `opcode`—4 characters of the command.
- `addr`—the address field is max 8 characters long, and contains the hexadecimal address value without '0x'.
- `length`—in case of a write command, this field indicates the size of the data to be sent, in hexadecimal, without '0x'.
- `not_used`—this field is not used for the moment.
- `rw`—indicates a read or write operation (00 for read, 01 for write).

Table 19-10. Simple Commands

Command	Full Name	Description
SEMB	SET_EXT_MEM_BOOT_EN	Set external memory boot enable
SKIF	SET_KEYS_IN_FUSE	Set keys in fuse
SPAI	SET_PAIRING_MODE	Set Pairing mode
SMDI	SET_MONITOR_DIS	Set monitor disable
SBSD	SET_BSCR_DIS	Set BSCR disable
SSBD	SET_SEC_DBG	Set secure debug
SJTD	SET_JTAG_DIS	Set JTAG disable fuse bit
CRST(1)	CHIP_RESET	Reset the chip

Note:

1. USB communication is cut when this command is executed by the chip.

Table 19-11. Read Version Command

Command	Full Name	Description
RVER	READ_ROM_VERSION	Read ROM code version

Table 19-12. Key Management Commands

Command	Full Name	Description
WCKY	WRITE_CUSTOMER_KEY	Send customer keys and initialization vector (640 bits)
WRHA	WRITE_RSA_HASH	Send RSA hash of root certificate public key

Note: Sending the commands in the table above automatically configures the corresponding values in the SIGN_MODE and EXPAND_MODE fields of the Boot Configuration Word.

Table 19-13. Applet Related Commands

Command	Full Name	Description
SAPT	SEND_APPLET	Send memory programming applet
SMBX	SEND_APPLET_MAILBOX	Send applet mailbox content

...........continued		
Command	**Full Name**	**Description**
RMBX	READ_APPLET_MAILBOX	Read applet mailbox content
EAPP	APPLET_EXECUTE	Execute apple
SFIL	SEND_FILE	Send file (to be programmed into an external memory)
RFIL	READ_FILE	Read file from an external memory

19.2.9.3 Monitor Answer

The Secure SAM-BA monitor answers any command with an acknowledge message formatted as follows:

____,________,________#[<`payload`>]

`opcode errcode length`

where:

- `opcode`—4 characters of the command
- `errcode`—value of the error code
- `length`—in cases where the monitor must send data to SAM-BA, this field shows how much data is in the payload, thus how much data SAM-BA has to read.
- `payload`—optional data, depends on the command.

Table 19-14. Command List

Command	Full Name	Description
CACK	SEND_CMD_ACK	Return the status of simple commands
ASTA	SEND_APPLET_STATUS	Return the status after execution of an applet
SVER	SEND_ROM_VERSION	Return ROM Code version

Table 19-15. Error Code

Error name	Hex value	Description
secCmdOK	0x000000000	Command OK
secCmdTooLong	0xFFFFFFFF	Whole command size too long
secCmdOpcodeSizeErr	0xFFFFFFFE	Opcode size too long
secCmdAddrSizeErr	0xFFFFFFFD	Address field size too long
secCmdLenSizeErr	0xFFFFFFFC	Length field size too long
secCmdRWSizeErr	0xFFFFFFFA	RW field size too long
secCmdOpcodeUnknown	0xFFFFFFF9	Unknown command op-code
secCmdCustKeyLengthErr	0xFFFFFFF8	Customer key payload message size error
secCmdCustKeyNotWritten	0xFFFFFFF7	Customer key has not been written
secCmdCustKeyAlreadyWritten	0xFFFFFFF6	A customer key has already been written in fuses
secCmdCmacErr	0xFFFFFFF5	Message CMAC error
secCmdDecryptErr	0xFFFFFFF4	Error during message decryption
secCmdKeyDerivErr	0xFFFFFFF3	Error during key derivation
secCmdRsaHashAlreadyWritten	0xFFFFFFEE	An RSA hash has already been written in fuses
secCmdFuseWriteErr	0xFFFFFFED	Error while writing a fuse bit

...........continued

Error name	Hex value	Description
secCmdExpandModeAlreadyWritten	0xFFFFFFEC	Field EXPAND_MODE in the Boot Configuration word has already been programmed

19.2.9.4 Command Description

19.2.9.4.1 Read ROM Code Version

For this specific command, the Secure SAM-BA monitor replies with the SVER opcode, and indicates that the ROM code version string length is 30 bytes, so the external tool knows how many character should be received.

(PC to device) `>> RVER,0,0,0,00#`

(Device to PC) `<< SVER,00000000,00000030#v1.0...........Jul 26 2019.....011:14:22.......`

19.2.9.4.2 Write Customer Key

Sending the customer key to the chip involves sending a binary payload (the ciphered message containing the customer key). This payload must be sent by the tool after receiving the acknowledge from the monitor, showing how many bytes are expected.

(PC to device) `>> WCKY,0,70,0,01#`

(Device to PC) `<< CACK,00000000,00000070#`

(PC to device) `>> <key_file.cip>`

(Device to PC) `<< CACK,00000000,00000000#`

19.2.9.4.3 Programming External Memories

Programming external memories requires the use of applets, which are memory programming algorithms running in the target internal SRAM.

Applets permit ROM code size reduction, no hard-coded memory identification tables, and give the possibility to add support for several types of external memories.

Applet binaries are ciphered and bundled into the Secure SAM-BA tool. They cannot be modified by the user and only Microchip can provide them.

Depending on the memory and applet implementation, the number of commands available can differ from one applet to another.

The first thing to do is to send the ciphered applet to the target. This is done through the Send Applet command:

(Device to PC) `>> SAPT,0,9870,0,01#`

(Device to PC) `<< CACK,00000000,00009870#`

(PC to device) `>> <applet_binary.cip>`

(Device to PC) `<< CACK,00000000,00000000#`

In the example above, Secure SAM-BA requests sending an applet of size 0x9870 and the monitor acknowledges this. Secure SAM-BA sends the applet ciphered binary file (applet_binary.cip), and after checking the signature and deciphering the applet in SRAM, the monitor sends the status (0x0: successful).

Now that the applet is in SRAM, before executing its code, its mailbox must be filled. The mailbox is the 32-word buffer, at the beginning of the applet area, which allows exchanging commands and arguments with the applet.

To do so, the Send Applet Mailbox command must be issued.

The mailbox is not ciphered, and is automatically written at the correct address by the ROM code.

(PC to device) `>> SMBX,0,80,0,01#`

(Device to PC) `<< CACK,00000000,00000080#`

(PC to device) `>> <applet_init_mailbox.bin>`

(Device to PC) `<< CACK,00000000,00000000#`

Now the applet program can be run with the Execute Applet command:

(PC to device) `>> EAPP,0,0,0,00#`

(Device to PC) `<< ASTA,00000000,00000000#`

The monitor replies with a status of the applet execution (0x0: successful).

During this step, the ROM code has retrieved the information that will be used when receiving the next Write File command:

- A memory has been successfully detected/initialized
- The address and the size of the data buffer: where the data exchanged can be stored (the final binary to be programmed into the external memory can be bigger than the buffer size)

Now a Send File command can be issued, with the size:

(PC to device) `>> SFIL,0,5000,0,01#`

(Device to PC) `<< CACK,00000000,00004000#`

(PC to device) `>> <first 0x4000 bytes of the file to be programmed>`

(Device to PC) `<< CACK,00000000,00001000#`

(PC to device) `>> <next file chunk of 0x1000 bytes>`

(Device to PC) `<< CACK,00000000,00000000#`

19.3 Key Provisioning and Bootstrap Programming

This section provides some recommendations to ensure a good security level when configuring and using Secure Boot mode.

19.3.1 Configuring Secure Boot Mode

The recommended procedure to configure the Secure Boot mode, using the SAM-BA tool (available on www.microchip.com), is the following:

1. Write the Boot Configuration Packet, with the required boot settings and boot memory interface.
2. Set the Secure Boot mode.
3. Send the customer key.
4. Send the Root Certificate Hash (in case RSA signature is used).
5. Configure the boot memory interface.
6. Program the ciphered bootstrap.
7. Program the other application files.
8. **Disable the monitor to avoid any further access to the Secure monitor.**
9. **Lock the Boot Configuration Packet.**

Note: Keeping the Secure Monitor enabled in order to update the bootstrap on the field or in-house is not recommended.

⚠WARNING The Boot Configuration Packet must be written as the very first packet in OTP.

19.3.2 Bootstrap Development and Updates

This section provides information about the bootstrap and how to update the ciphered bootstrap on a system already provisioned and having the secure monitor disabled.

In order to protect the CBC ciphered bootstrap from known plain text attacks, it is strongly recommended to follow several common rules:

- Avoid compiling and linking “as is” source code that is publicly available.
- Apply secure software development basic principles (add custom parts, add random data, change objects order, etc.).
- Keep bootstrap sources (binary and ciphered versions) in safe places.
- Keep bootstrap as small and robust as possible to reduce number of updates.

19.3.2.1 Bootstrap Ciphering

Before update, the bootstrap must be ciphered in accordance with the Secure Boot mode selected.

When the Pairing mode is used, the bootstrap authentication code and ciphering differ for each device, because it is paired using the device unique ID. Remote update in the field is possible if the system can connect to a server. The system sends its chip ID to the server, and the server computes the new bootstrap that is sent back to the system.

In this case, it is highly recommended to not try any field update in this mode.

For other Secure Boot modes, the bootstrap can be prepared using the Secure SAM-BA cipher tool, and then deployed for an update on all devices.

Securing *.CIP files

Microchip’s Secure SAM-BA Cipher utility outputs encrypted *.CIP files for use in provisioning at final manufacturing. As these files contain both customer keys and firmware, the *.CIP files must be securely stored with restricted and controlled access.

19.3.2.2 Bootstrap Field Update

19.3.2.2.1 Case 1: SAM-BA Monitor is disabled (the “MON_DIS” word is filled in the Boot Configuration Packet)

In this case, the bootstrap itself or the customer application must be in charge to update the bootstrap, by writing the new bootstrap in the boot memory in replacement of the existing one.

To do that, the program in charge to do the update must be able to:

- retrieve the new ciphered bootstrap,
- erase the existing one,
- program the new one.

Important: The last two operations must not be interrupted, otherwise the boot process could be broken.

19.3.2.2.2 Case 2: Secure SAM-BA Monitor is still available (the “MON_DIS” word is zeroed in the Boot Configuration Packet - not recommended for systems in production)

1. Disable the access to the external boot memory (Chip Select pin, Card Detect Pin, etc., depending on the chosen boot memory) to prevent its access by the ROM code on the next reboot.
2. Connect a USB cable on the USB device port, or on the UART used as SAM-BA Monitor console terminal.
3. Reset the chip.
4. Enable the access to the external boot memory.
5. Program the bootstrap using the SAM-BA tool.
6. Lock the Boot Configuration Packet.

19.3.3 Monitor Disabling by the Bootstrap

It is highly recommended to fill this word and lock the Boot Configuration Packet at the production stage, but if none of those conditions are met, writing the “MON_DIS” word can be performed by the bootstrap or by the customer application.

To write the “MON_DIS” word and lock it, an update of the Boot Configuration Packet in the OTP memory is required. In the section OTP Memory Controller (OTPC), refer to:

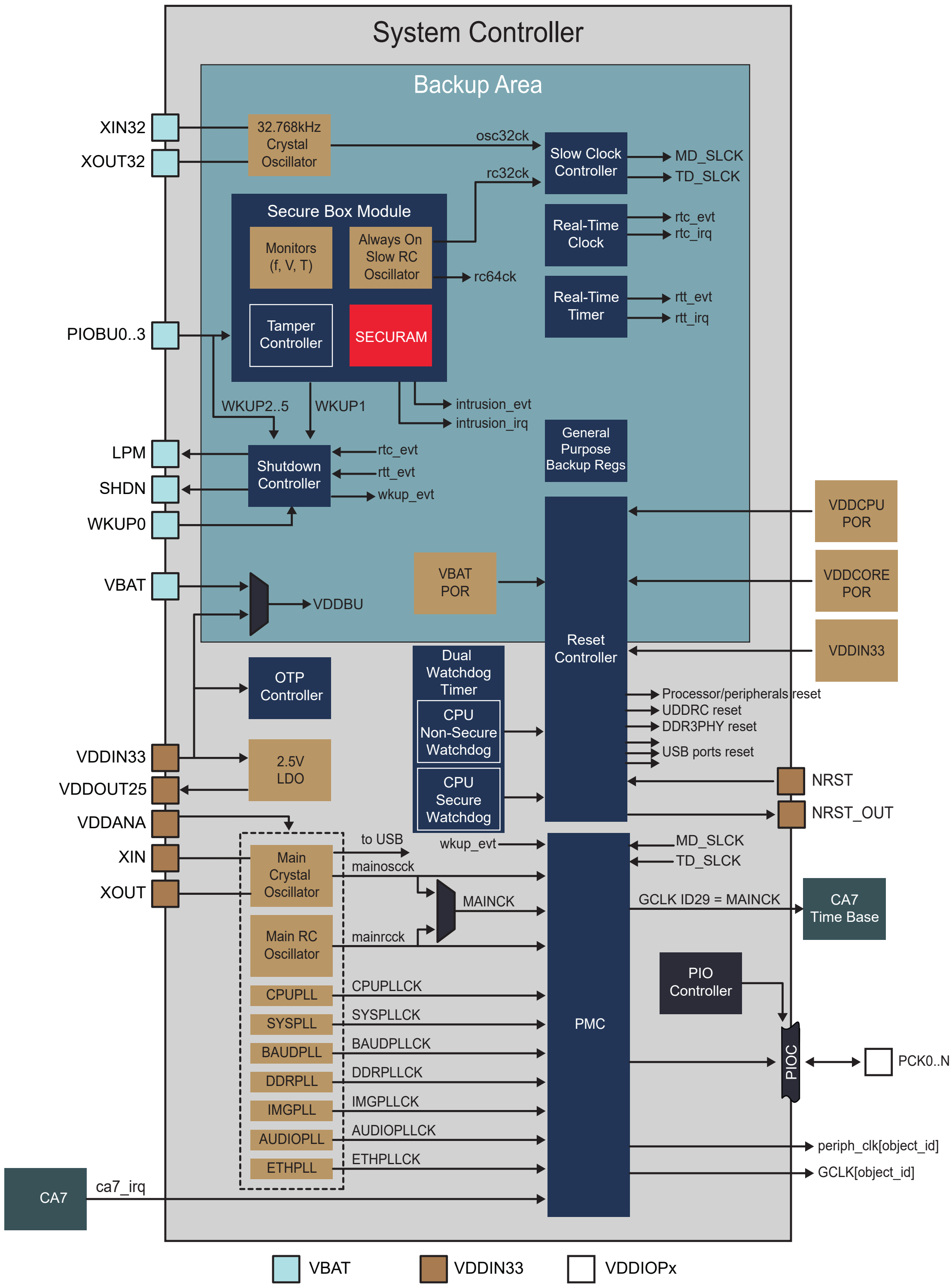
- Updating an Existing Packet from the User Interface

- Locking a Packet

20. SYSTEM CONTROLLER SUBSYSTEM

20.1 Block Diagram

Figure 20-1. System Controller Block Diagram



20.2 Components

- Peripherals powered by VDDBU (either VBAT or VDDIN33):
  - System Controller Write Protection (SYSCWP)
  - General Purpose Backup Registers (GPBR)
  - Reset Controller (RSTC)
  - Real-time Timer (RTT)
  - Real-time Clock (RTC)
  - Shutdown Controller (SHDWC)
  - Slow Clock Controller (SCKC)
  - Special Function Registers Backup (SFRBU)
  - Security Module (SECUMOD)
- Peripherals powered by VDDCORE:
  - Dual Watchdog Timer (DWDT)
  - Chip Identifier (CHIPID)
  - OTP Memory Controller (OTPC)
  - Special Function Registers (SFR)
  - Clock Generator and Power Management Controller (PMC)
  - Parallel Input/Output Controller (PIO)
  - 64-bit Periodic Interval Timer (PIT64B)

20.3 Product Dependencies

20.3.1 Clocks

All system controller peripherals, except the 6x PIT64B instances, are part of the CPU System and Security (CSS) matrix that runs on MCK0.

Note: The MCK0 frequency is directly related to the CPU clock, so any change on the CPU clock impacts MCK0.

The system controller peripherals are always on, except:

- Parallel Input/Output Controllers (PIOs) that have one clock control (PIOA) for all PIO controllers, clocked with MCK0
- Special Function Registers (SFR)

PIT64B0 to PIT64B5 are located on the APB Client (APS) matrix, clocked with MCK1.

In addition, PIT64B0 to PIT64B5 feature a GCLK input for flexibility. CA7 Base Time (GCLK ID29), defined as MAINCK, is automatically started at reset.

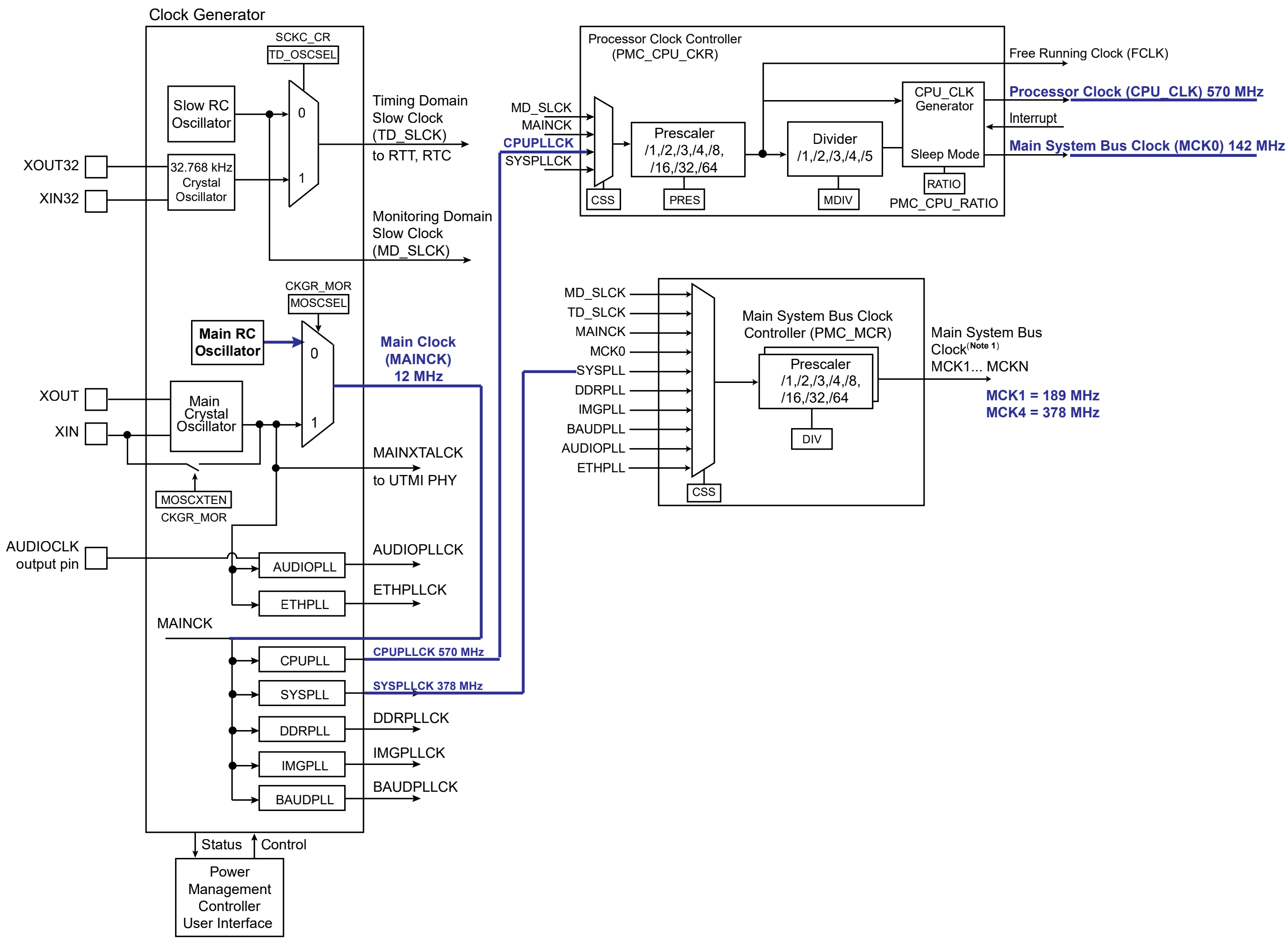
20.3.1.1 Changing System Frequencies

After ROM code execution, the system clock is set as follows:

- MAINCK is fed by the main RC oscillator i.e., 12 MHz.
- CPUPLL is set to 570 MHz.
- MCK0 is set to CPUPLL/4 i.e., 142 MHz.
- SYSPLL is set to 378 MHz.
- MCK1 is set to SYSPLL/2 i.e., 189 MHz.
- MCK4 is set to SYSPLL i.e., 378 MHz.

This is illustrated in the following figure.

Figure 20-2. PMC Rom Code Configuration



To avoid any overclocking during system clock modification, it is recommended to change system frequencies in two steps.

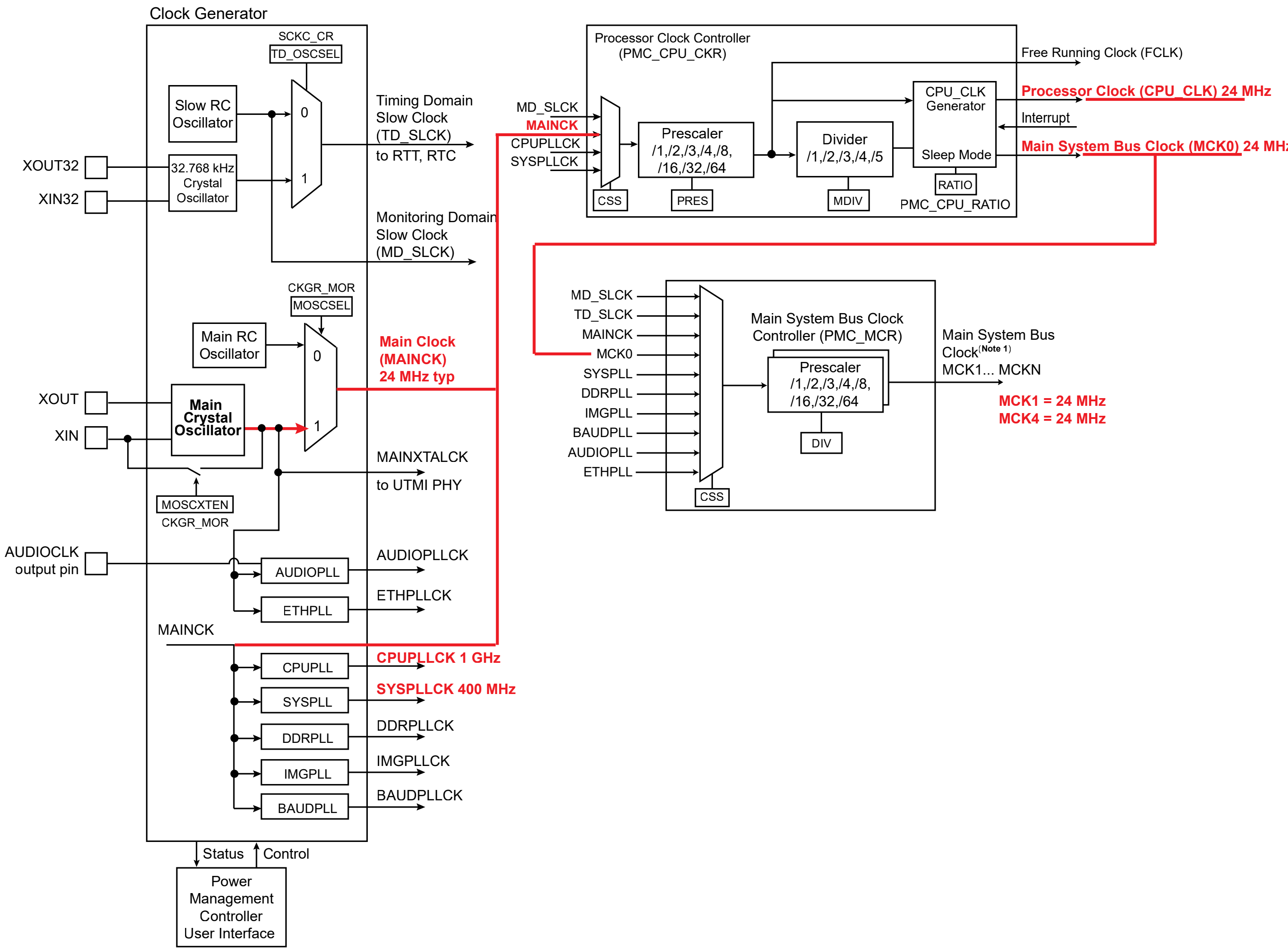
The first step leads to a known and basic intermediate state where all clocks are in a low-frequency range.

For example, the following sequence can be performed:

1. Set MCK0 to MAINCK, i.e., 12 MHz.
2. Set MCK1 to MCK0 i.e., 12 MHz.
3. Set MCK4 to MCK0 i.e., 12 MHz.
4. Set MAINCK to Crystal Oscillator, typically 24 MHz. MCK0, MCK1 and MCK4 then run at 24 MHz.
5. Set CPUPLL to 1 GHz.
6. Set SYSPLL to 400 MHz.

This intermediate state is illustrated in the following figure.

Figure 20-3. PMC Intermediate Configuration



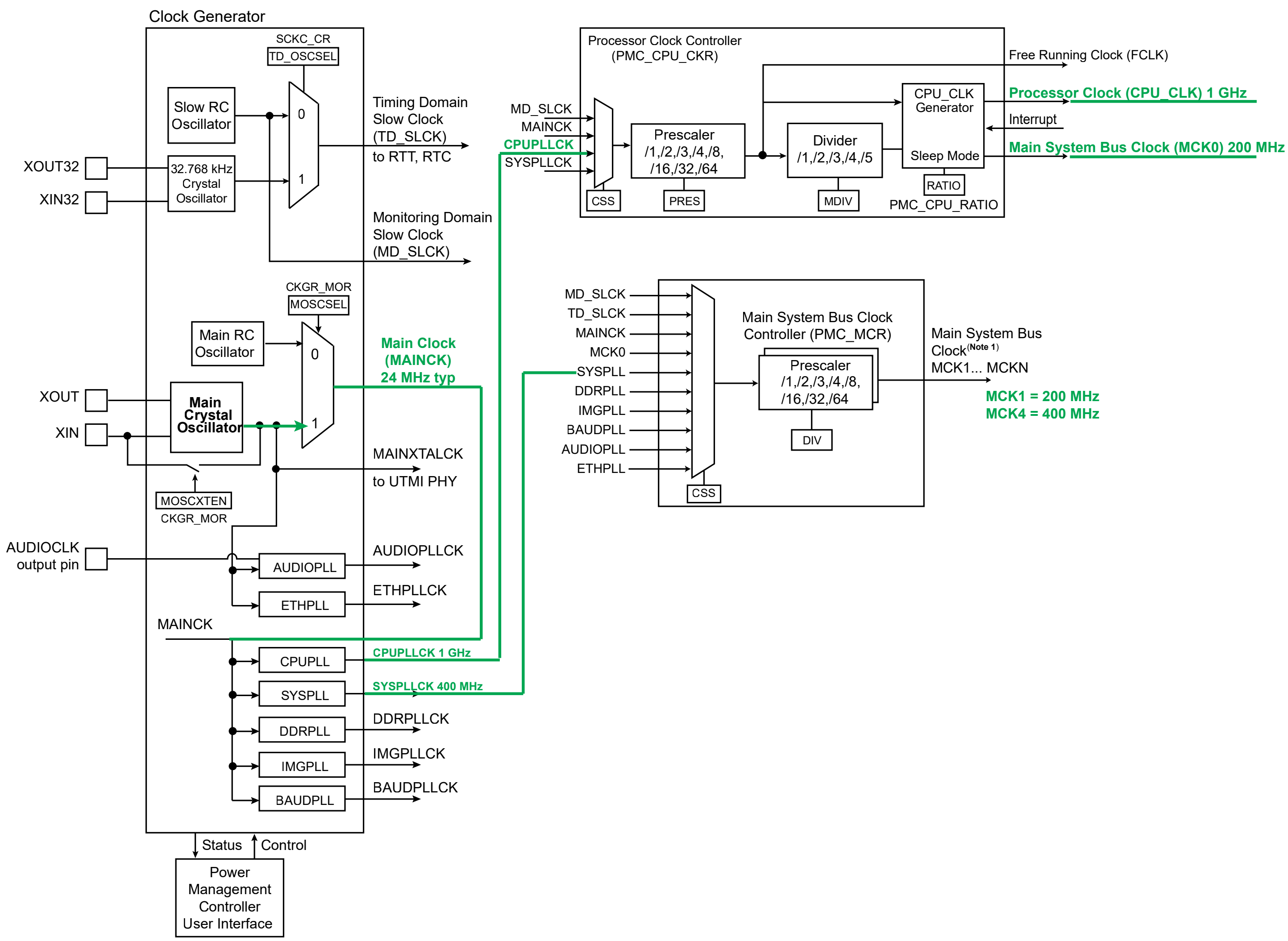
The second step leads to the final expected state.

Perform the following sequence:

1. Set MCK0 to CPUPLL/5 i.e., 200 MHz.
2. Set MCK1 to SYSPLL/2 i.e., 200 MHz.
3. Set MCK4 to SYSPLL i.e., 400 MHz.

The final state is illustrated in the following figure.

Figure 20-4. PMC Final Configuration



20.3.2 Interrupts

Each peripheral has its own GIC interrupt ID, except CHIP_ID, SFRBU, SYSCWP and GPBR, that have no interrupt line.

DWDT has two separated interrupts: one programmable secure and one never secure.

20.3.3 Reset

System controller peripherals are connected to the processor and peripherals reset line.

20.3.4 I/Os

For the applicable I/O type and power supply, refer to the table Pin Description.

20.3.5 Power Supplies

The 32.768 kHz crystal oscillator, the slow RC oscillator and the secure backup SRAM (SECURAM) are powered by VDDBU.

The main crystal oscillator, the main RC oscillator and all PLLs are powered by VDDANA.

The VDDOUT25 2.5V LDO regulator output must be connected to VDDANA and VDDDHY to fulfill the electrical requirements of these power inputs.

20.4 Special Functions in SFR/SFRBU

The backup power switch control is located in the SFRBU_PSWBU register.

The 2.5V LDO regulator control is located in the SFRBU_LDO25CR register.

21. System Controller Write Protection (SYSCWP)

21.1 Functional Description

21.1.1 System Controller Peripheral Mapping

Table 21-1. System Controller Peripheral Mapping

Offset	System Controller Peripheral	Name
0x000-0x00C	Reset Controller	RSTC
0x010-0x01C	Shutdown Controller	SHDWC
0x020-0x03C	Real Time Timer	RTT
0x050-0x05C	Slow Clock Controller	SCKC
0x060-0x0A4	General Purpose Backup Registers	GPBR
0x0A8-0xD8	Real Time Clock	RTC
0x0DC	Write Protection Mode Register	SYSC_WPMR
0x0E0	Write Protection Status Register	SYSC_WPSR
0x100-0x16C	Real Time Clock (tamper control and timestamp)	RTC
0x180-0x1AC	Watchdog Timer	WDT

21.1.2 Register Write Protection

To prevent any single software error from modifying the configuration of the Reset Controller (RSTC), Shutdown Controller (SHDWC), Real-time Timer (RTT), Slow Clock Controller (SCKC), General Purpose Backup Register (GPBR), Real-time Clock (RTC) and Watchdog Timer (WDT), some registers of these peripherals can be write-protected by setting the WPEN and/or WPITEN bits in the System Controller Write Protection Mode register (SYSC_WPMR).

Note: The WDT embeds additional write protection mechanisms.

When write protection is enabled, any attempt to write these registers is reported in the System Controller Write Protection Status register (SYSC_WPSR).

The following registers can be write-protected when SYSC_WPMR.WPEN=1:

- RSTC Mode Register
- SHDWC Mode Register
- SHDWC Wakeup Inputs Register
- SCKC Configuration Register
- RTC Control Register
- RTC Mode Register
- RTC Time Alarm Register
- RTC Calendar Alarm Register
- RTT Mode Register
- RTT Alarm Register
- RTT Modulo Selection Register
- GPBR Full Clear Register
- GPBR Registers

The following registers can be write-protected when SYSC_WPMR.WPITEN=1:

- RTC Interrupt Enable Register
- RTC Interrupt Disable Register

21.2 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SYSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x04	SYSC_WPSR	31:24								
		23:16								
		15:8	WVSRC[7:0]							
		7:0								WPVS

21.2.1 SYSC Write Protection Mode Register

Name: SYSC_WPMR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535943	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 1 – WPITEN Write Protection RTC Interrupt Enable

Value	Description
0	Disables the write protection of the RTC_IER/RTC_IDR configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the RTC_IER/RTC_IDR configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

21.2.2 SYSC Write Protection Status Register

Name: SYSC_WPSR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WVSRC[7:0] Write Violation Source
When bit WPVS is equal to 1, the field WVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Register Violation Status
WDT_CR, WDT_MR, RTT_MODR, RTC_IDR and RTC_IER can be write-protected but WPVS does not report any violation for these registers.

Value	Description
0	No write protection violation has occurred since the last read of SYSC_WPSR.
1	A write protection violation has occurred since the last read of SYSC_WPSR. The associated violation is reported into field WVSRC.

22. General Purpose Backup Registers (GPBR)

22.1 Description

The System Controller embeds 64 bits of General Purpose Backup registers organized as 2 32-bit registers.

SYS_GPBR0 to SYS_GPBR1 can be individually (each 32-bit part-select) read- and write-protected by configuring GPBR_MR. This register is write-once, which means that once it has been configured, the read or write protection is available until the loss of VBAT.

22.2 Embedded Characteristics

- 64 bits of General Purpose Backup Registers
- Read and Write Protection for SYS_GPBR0 to SYS_GPBR1

22.3 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	GPBR_MR	31:24								
		23:16							GPBRRP1	GPBRRP0
		15:8								
		7:0							GPBRWP1	GPBRWP0
0x04	GPBR_FCLR	31:24								
		23:16								
		15:8								
		7:0								FCLR
0x08	SYS_GPBR0	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x0C	SYS_GPBR1	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							

22.3.1 GPBR Mode Register

Name: GPBR_MR
Offset: 0x0
Reset: 0x00000000
Property: Read/Write-Once

This register is write-once. All bits are cleared at first power-up and on each loss of VBAT.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							GPBRRP1	GPBRRP0
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							GPBRWP1	GPBRWP0
Access							R/W	R/W
Reset							0	0

Bits 16, 17 – GPBRRPx GPBRx Read Protection

Value	Description
0	The content of the corresponding GPBR register (32-bit part-select) can be read.
1	The corresponding GPBR register (32-bit part-select) always returns zero when read.

Bits 0, 1 – GPBRWPx GPBRx Write Protection

Value	Description
0	The corresponding GPBR register (32-bit part-select) can be written.
1	The corresponding GPBR register (32-bit part-select) is write-protected.

22.3.2 GPBR Full Clear Register

Name: GPBR_FCLR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

All bits are cleared at first power-up and on each loss of VBAT.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FCLR
Access								R/W
Reset								0

Bit 0 – FCLR Full Clear Enable

Value	Description
0	SYS_GPBR0 to SYS_GPBR1 are immediately cleared in case of fast wake-up pin tamper event.
1	All SYS_GPBRx are immediately cleared in case of fast wake-up pin tamper event.

22.3.3 General Purpose Backup Register x [x=0..1]

Name: SYS_GPBRx
Offset: 0x08 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

These registers are reset at first power-up and on each loss of VBAT.

Bit	31	30	29	28	27	26	25	24
	GPBR_VALUE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GPBR_VALUE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	GPBR_VALUE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GPBR_VALUE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GPBR_VALUE[31:0] Value of SYS_GPBRx
Note: If an event has been detected on WKUP pins enabled for tamper event detection in the RTC, it is not possible to write to SYS_GPBRx as long as the event has not been cleared.

23. Dual Watchdog Timer (DWDT)

23.1 Description

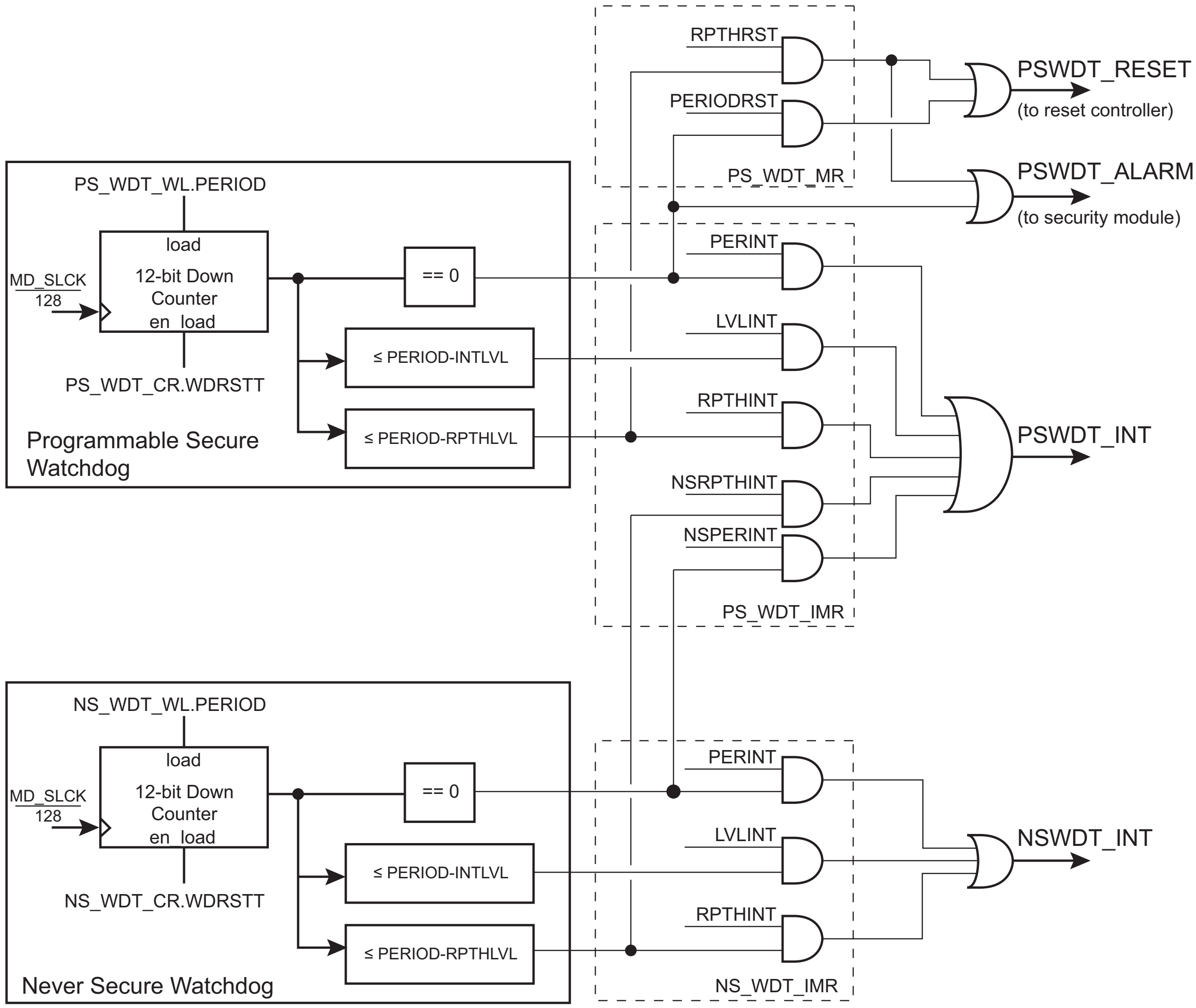
The Dual Watchdog Timer (DWDT) is used to prevent system lock-up if the software becomes trapped in a deadlock. It features two 12-bit down counters that allow two watchdog periods of up to 16 seconds (Monitoring Domain Slow Clock (MD_SLCK) around 32 kHz). In addition, it can be stopped while the processor is in Debug mode or Sleep mode (Idle mode).

23.2 Embedded Characteristics

- 12-bit Key-Protected Programmable Counter
- Watchdog Clock is Independent from Processor Clock
- Provides Reset or Interrupt Signals to the System
- Counter May be Stopped while the Processor is in Debug State or in Idle Mode

23.3 Block Diagram

Figure 23-1. DWDT Block Diagram



23.4 Functional Description

23.4.1 Configuration

The DWDT is used to prevent both secure and never secure system lock-up if the software becomes trapped in a deadlock. It is supplied with VDDCORE. It restarts with initial values on processor reset. It embeds two watchdogs:

- NSWDT–Never Secure Watchdog
- PSWDT–Programmable Secure Watchdog

Each watchdog is built around a 12-bit down counter, which is loaded with the value defined in the field PERIOD of the Window Level register (NS_WDT_WL or PS_WDT_WL). These registers use MD_SLCK divided by 128 to establish the maximum watchdog period to be 16 seconds (with a typical MD_SLCK of 32.768 kHz).

For each watchdog, the following parameters can be defined:

- PERIOD (Watchdog Period)–Load value of the down counter. Once the down counter reaches 0, a watchdog event is generated. For the PSWDT, a watchdog event leads to either a secure interrupt (if PS_WDT_IMR.PERINT is set to '1') or a reset (if PS_WDT_MR.PERIODRST is set to '1'). For the NSWDT, a watchdog event leads to either a secure interrupt (if PS_WDT_IMR.NSPERINT is set to '1') and/or a never secure interrupt (if NS_WDT_IMR.PERINT is set to '1').

- RPTH (Repeat Threshold)–A watchdog restart done before the repeat threshold is elapsed leads to a repeat violation. For the PSWDT, a repeat violation leads to either a secure interrupt (if PS_WDT_IMR.RPTHINT is set to '1') or a reset (if PS_WDT_MR.RPTHRST is set to '1'). For the NSWDT, a repeat violation leads to either a secure interrupt (if PS_WDT_IMR.NSRPTHINT is set to '1') or a never secure interrupt (if NS_WDT_IMR.RPTHINT is set to '1').
- LVLTH (Interrupt Threshold)–Threshold after which an interrupt is generated. For the PSWDT, a repeat violation leads to a secure interrupt (if PS_WDT_IMR.LVLINT is set to '1'). For the NSWDT, a repeat violation leads to a never secure interrupt (if NS_WDT_IMR.LVLINT is set to '1').

After a processor reset, the value of PERIOD is 0xFFF, corresponding to the maximum value of the counter with the external reset generation enabled (field PERIODRST at 1 after a backup reset). This means that watchdogs are running at reset, i.e., at power-up. The user can either disable the WDT by setting bit PS_WDT_MR.WDDIS and/or NS_WDT_MR.WDDIS to '1' or reprogram the WDTs to meet the maximum watchdog period the application requires.

The NSWDT and the PSWDT embed securities to avoid programming out of range values. The following inequality must always be respected, otherwise the configuration is cancelled:

$$\mathrm{RPTH} \leq \mathrm{LVLTH} < \mathrm{PERIOD}$$

Moreover, the PSWDT has the possibility to control the range of operation of the NSWDT. It can limit the period, the repeat threshold and the interrupt level of the NSWDT by programming NS_WDT_LVLLIM, NS_WDT_RLIM and NS_WDT_PLIM.

23.4.2 Watchdog Reload

In normal operation, the user reloads the watchdog at regular intervals before the down counter reaches 0, by setting PS_WDT_CR.WDRSTT or NS_WDT_CR.WDRSTT with the right KEY field. The watchdog counter is then immediately reloaded with the PERIOD value and restarted, and the MD_SLCK 128 divider is reset and restarted. Writing PS_WDT_CR or NS_WDT_CR with a wrong key has no effect.

Writing PS_WDT_MR, PS_WDT_WLR or PS_WDT_ILR immediately reloads the counter from PERIOD and restarts the MD_SLCK 128 divider of PSWDT.

Writing NS_WDT_MR, NS_WDT_WLR or NS_WDT_ILR immediately reloads the counter from PERIOD and restarts the MD_SLCK 128 divider of NSWDT.

23.4.3 Watchdog Lock

PS_WDT_MR and NS_WDT_MR can be written until a LOCKMR command is issued in the corresponding PS_WDT_CR or NS_WDT_CR. Once locked, only a processor reset resets PS_WDT_MR and NS_WDT_MR. As long as a WDT is not locked, writing NS_WDT_MR or PS_WDT_MR automatically reloads the corresponding Watchdog timer with the newly programmed mode parameters.

If the watchdog is restarted by writing into the corresponding Control register (PS_WDT_CR or NS_WDT_CR), the corresponding PS_WDT_MR or NS_WDT_MR must not be programmed during a period of time of three MD_SLCK periods following the PS_WDT_CR or NS_WDT_CR write access.

23.4.4 Repeat Threshold

A repeat threshold can be defined for each watchdog in order to protect against dead-locks that would repeatedly restart the watchdog. PS_WDT_WL.RPTH and NS_WDT_WL.RPTH define the minimum number of cycles to wait after a watchdog restart before the watchdog can be restart again. If a watchdog restart occurs before this limit is reached, a repeat threshold failure is asserted and the PS_WDT_ISR.RPTHINT and/or the NS_WDT_ISR.RPTHINT bit are/is set to one. This feature can be disabled by programming a null RPTH value. In such a configuration, restarting the Watchdog Timer is permitted in the whole range [0; PERIOD] and does not generate an error. This is the default configuration on reset (RPTH is null).

23.4.5 Watchdog Reset Order

Only the Programmable Secure watchdog (PSWDT) can send a Watchdog Reset Order to the Reset Controller. If the down counter reaches 0 or if the watchdog is restarted before reaching the RPTH threshold limit, a Watchdog Reset Order is sent to the Reset Controller if PS_WDT_MR.PERIODRST and/or PS_WDT_MR.RPTHRST are set to '1'.

23.4.6 Watchdog Interrupt

If PS_WDT_IMR.RPTHINT is set to '1' and a repeat threshold violation occurs in the PSWDT, a programmable secure interrupt is generated.

If NS_WDT_IMR.RPTHINT is set to '1' and a repeat threshold violation occurs in the NSWDT, a never secure interrupt is generated.

If PS_WDT_IMR.NSRPTHINT is set to '1' and a repeat threshold violation occurs in the NSWDT, a programmable secure interrupt is generated.

If a reset is generated or if PS_WDT_SR or NS_WDT_SR is read, the status bits are reset, the interrupt is cleared, and the “wdt_fault” signal to the reset controller is deasserted.

23.4.7 Security Module

Both watchdogs are connected to the security module. If a down counter reaches 0, an alarm is systematically sent to the security module (whatever the configuration of interrupts and resets).

If the PSWDT is restarted before the RPTH threshold, an alarm is sent to the security module only if PS_WDT_MR.RPTHRST is set to ‘1’. Nevertheless, this does not cancel the reset operation.

23.4.8 Watchdog Halt

While the processor is in Debug state or in Sleep mode (including ULP mode 1 and 2), the counter may be stopped depending on the value programmed for the bits WDIDLEHLT and WDDBGHLT in PSWDT_MR or NSWDT_MR.

23.4.9 Timing Diagrams

Figure 23-2. Never Secure Watchdog Timing Diagram

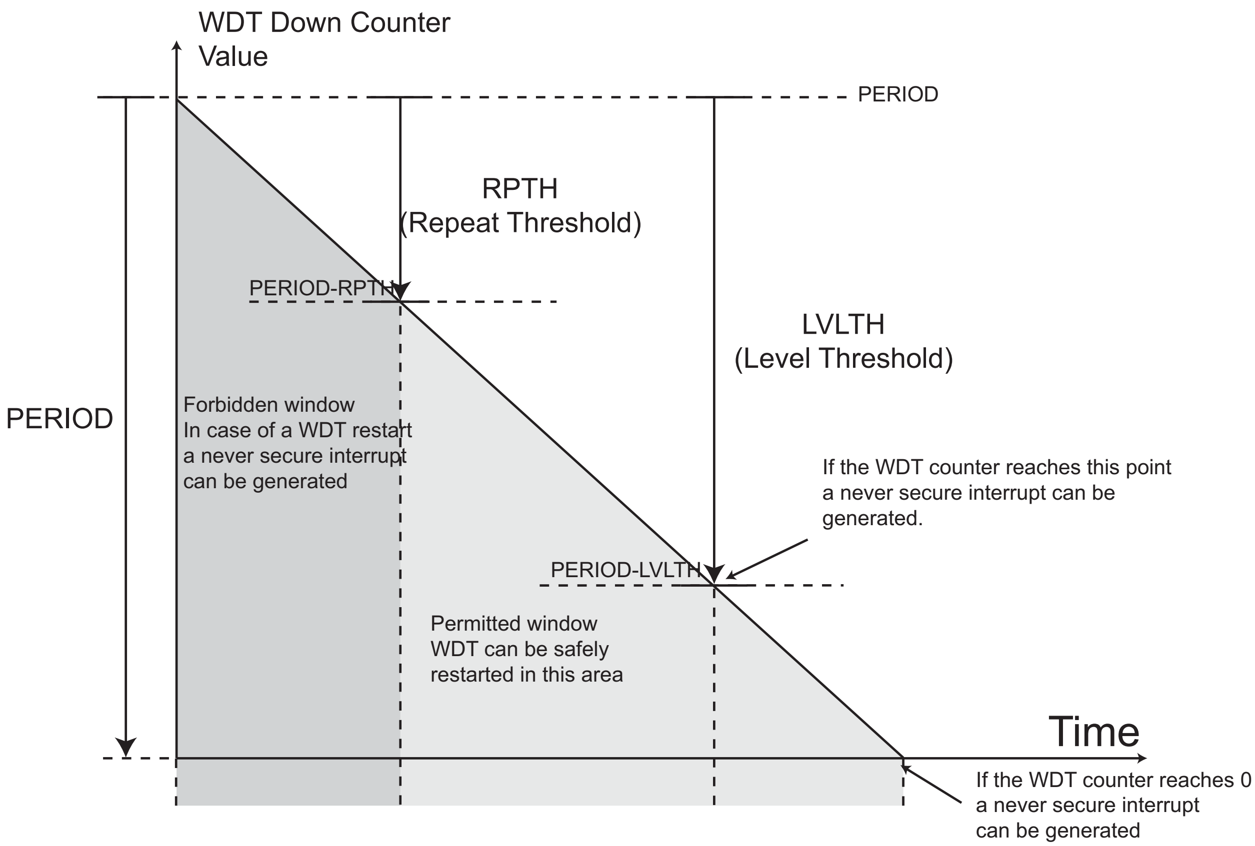
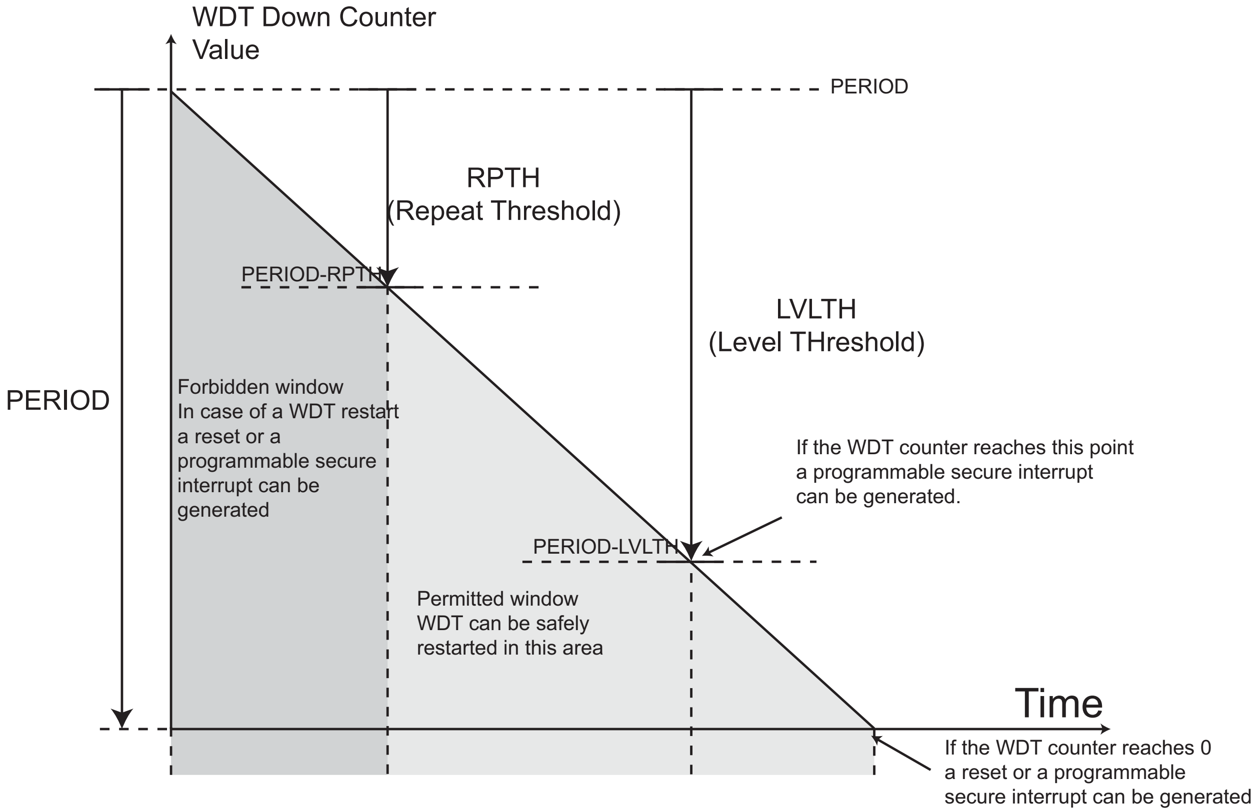


Figure 23-3. Programmable Secure Watchdog Timing Diagram



23.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	NS_WDT_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x04	NS_WDT_MR	31:24			WDIDLEHLT	WDDBGHLT				
		23:16								
		15:8				WDDIS				
		7:0								
0x08	NS_WDT_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x0C	NS_WDT_WL	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x10	NS_WDT_IL	31:24								
		23:16								
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							
0x14	NS_WDT_IER	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x18	NS_WDT_IDR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x1C	NS_WDT_ISR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x20	NS_WDT_IMR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x24 ... 0x0FFF	Reserved									
0x1000	Reserved for SYSC	31:24								
		23:16								
		15:8								
		7:0								
...										
0x117C	Reserved for SYSC	31:24								
		23:16								
		15:8								
		7:0								
0x1180	PS_WDT_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x1184	PS_WDT_MR	31:24			WDIDLEHLT	WDDBGHLT				
		23:16								
		15:8				WDDIS				
		7:0			RPTHRST	PERIODRST				

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1188	PS_WDT_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x118C	PS_WDT_WL	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x1190	PS_WDT_IL	31:24								
		23:16								
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							
0x1194	PS_WDT_IER	31:24								
		23:16								
		15:8								
		7:0				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
0x1198	PS_WDT_IDR	31:24								
		23:16								
		15:8								
		7:0				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
0x119C	PS_WDT_ISR	31:24								
		23:16								
		15:8								
		7:0				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
0x11A0	PS_WDT_IMR	31:24								
		23:16								
		15:8								
		7:0				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
0x11A4	NS_WDT_LVLLIM	31:24					LVLMAX[11:8]			
		23:16	LVLMAX[7:0]							
		15:8					LVLMIN[11:8]			
		7:0	LVLMIN[7:0]							
0x11A8	NS_WDT_RLIM	31:24					RPTHMAX[11:8]			
		23:16	RPTHMAX[7:0]							
		15:8					RPTHMIN[11:8]			
		7:0	RPTHMIN[7:0]							
0x11AC	NS_WDT_PLIM	31:24					PERMAX[11:8]			
		23:16	PERMAX[7:0]							
		15:8					PERMIN[11:8]			
		7:0	PERMIN[7:0]							

23.5.1 DWDT Never Secure Watchdog Timer Control Register

Name: NS_WDT_CR
Offset: 0x00
Reset: –
Property: Write-only

Note: The NS_WDT_CR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in NS_WDT_CR. Any modification causes the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LOCKMR				WDRSTT
Access				W				W
Reset				–				–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 4 – LOCKMR Lock Mode Register Write Access

Value	Description
0	No effect.
1	Locks the configuration registers if KEY is written to 0xA5. Write accesses to NS_WDT_MR, NS_WDT_WL and NS_WDT_IL have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the watchdog if KEY is written to 0xA5.

23.5.2 DWDT Never Secure Watchdog Timer Mode Register

Name: NS_WDT_MR
Offset: 0x04
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in NS_WDT_CR (unlocked on hardware reset).

The NS_WDT_MR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in NS_WDT_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDIDLEHLT	WDDBGHLT				
Access			R/W	R/W				
Reset			0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WDDIS				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 29 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The watchdog runs when the system is in Idle state.
1	The watchdog stops when the system is in Idle state.

Bit 28 – WDDBGHLT Watchdog Debug Halt

Value	Description
0	The watchdog runs when the processor is in Debug state.
1	The watchdog stops when the processor is in Debug state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

23.5.3 DWDT Never Secure Watchdog Timer Value Register

Name: NS_WDT_VR
Offset: 0x08
Reset: 0x00000FFF
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNTER[11:8]			
Access					R	R	R	R
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 11:0 – COUNTER[11:0] Watchdog Down Counter Value
Current value of the watchdog down counter.
Due to the asynchronous operation of the DWDT with respect to the rest of the chip, to be certain that the value read in this is valid and stable, it is necessary to read this register twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

23.5.4 DWDT Never Secure Watchdog Timer Window Level Register

Name: NS_WDT_WL
Offset: 0x0C
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RPTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PERIOD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:16 – RPTH[11:0] Repeat Threshold
Defines the period before which a watchdog restart generates a never secure and/or a programmable secure interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the Watchdog generates a never secure and/or a programmable secure interrupt.

23.5.5 DWDT Never Secure Watchdog Timer Interrupt Level Register

Name: NS_WDT_IL
Offset: 0x10
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					LVLTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	LVLTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 11:0 – LVLTH[11:0] Level Threshold
Defines the period after which the Watchdog generates a never secure interrupt.

23.5.6 DWDT Never Secure Watchdog Interrupt Enable Register

Name: NS_WDT_IER
Offset: 0x14
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Value	Description
0	No effect.
1	The never secure interrupt threshold failure interrupt is enabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Enable

Value	Description
0	No effect.
1	The never secure repeat threshold failure interrupt is enabled.

Bit 0 – PERINT Period Interrupt Enable

Value	Description
0	No effect.
1	The never secure period failure interrupt is enabled.

23.5.7 DWDT Never Secure Watchdog Interrupt Disable Register

Name: NS_WDT_IDR
Offset: 0x18
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Value	Description
0	No effect.
1	The never secure interrupt threshold failure interrupt is disabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Disable

Value	Description
0	No effect.
1	The never secure repeat threshold failure interrupt is disabled.

Bit 0 – PERINT Period Interrupt Disable

Value	Description
0	No effect.
1	The never secure period failure interrupt is disabled.

23.5.8 DWDT Never Secure Watchdog Interrupt Status Register

Name: NS_WDT_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						R	R	R
Reset						0	0	0

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Value	Description
0	No level threshold failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.
1	At least one level threshold failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.

Bit 1 – RPTHINT Repeat Threshold Interrupt Enable

Value	Description
0	No repeat threshold failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.
1	At least one repeat threshold failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.

Bit 0 – PERINT Period Interrupt Status

Value	Description
0	No period failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.
1	At least one period failure has occurred in the never secure watchdog since the last read of NS_WDT_ISR.

23.5.9 DWDT Never Secure Watchdog Interrupt Mask Register

Name: NS_WDT_IMR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						R	R	R
Reset						0	0	0

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Value	Description
0	Interrupt on LVLINT is disabled.
1	Interrupt on LVLINT is enabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Mask

Value	Description
0	Interrupt on RPTHINT is disabled.
1	Interrupt on RPTHINT is enabled.

Bit 0 – PERINT Period Interrupt Mask

Value	Description
0	Interrupt on PERINT is disabled.
1	Interrupt on PERINT is enabled.

23.5.10 DWDT Programmable Secure Watchdog Timer Control Register

Name: PS_WDT_CR
Offset: 0x1180
Reset: –
Property: Write-only

Register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in PS_WDT_CR. Any modification causes the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LOCKMR				WDRSTT
Access				W				W
Reset				–				–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 4 – LOCKMR Lock Mode Register Write Access

Value	Description
0	No effect.
1	Locks the configuration registers if KEY is written to 0xA5. Write accesses to PS_WDT_MR, PS_WDT_WL and PS_WDT_IL have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the watchdog if KEY is written to 0xA5.

23.5.11 DWDT Programmable Secure Watchdog Timer Mode Register

Name: PS_WDT_MR
Offset: 0x1184
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in PS_WDT_CR (unlocked on hardware reset).

The PS_WDT_MR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in PS_WDT_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDIDLEHLT	WDDBGHLT				
Access			R/W	R/W				
Reset			0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WDDIS				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
			RPTHRST	PERIODRST				
Access			R/W	R/W				
Reset			1	1				

Bit 29 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The watchdog runs when the system is in Idle state.
1	The watchdog stops when the system is in Idle state.

Bit 28 – WDDBGHLT Watchdog Debug Halt

Value	Description
0	The watchdog runs when the processor is in Debug state.
1	The watchdog stops when the processor is in Debug state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

Bit 5 – RPTHRST Repeat Threshold Reset

Value	Description
0	No reset is generated if the watchdog is restarted before the RPTH threshold
1	A reset is generated if the watchdog is restarted before the RPTH threshold

Bit 4 – PERIODRST Period Reset

Value	Description
0	No reset is generated if the watchdog down counter reaches 0

Value	Description
1	A reset is generated once the watchdog down counter reaches 0

23.5.12 DWDT Programmable Secure Watchdog Timer Value Register

Name: PS_WDT_VR
Offset: 0x1188
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNTER[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – COUNTER[11:0] Watchdog Down Counter Value
Current value of the watchdog down counter.
Due to the asynchronous operation of the DWDT with respect to the rest of the chip, to be certain that the value read in this is valid and stable, it is necessary to read this register twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

23.5.13 DWDT Programmable Secure Watchdog Timer Window Level Register

Name: PS_WDT_WL
Offset: 0x118C
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RPTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PERIOD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:16 – RPTH[11:0] Repeat Threshold
Defines the period before which a watchdog restart generates a programmable secure and/or a programmable secure interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the watchdog generates a programmable secure and/or a programmable secure interrupt.

23.5.14 DWDT Programmable Secure Watchdog Timer Interrupt Level Register

Name: PS_WDT_IL
Offset: 0x1190
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					LVLTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	LVLTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 11:0 – LVLTH[11:0] Level Threshold
Defines the period after which the Watchdog generates a Programmable secure interrupt.

23.5.15 DWDT Programmable Secure Watchdog Interrupt Enable Register

Name: PS_WDT_IER
Offset: 0x1194
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 4 – NSRPTHINT Never Secure Repeat Threshold Interrupt Enable

Value	Description
0	No effect.
1	A never secure repeat threshold failure generates an interrupt in the programmable secure area.

Bit 3 – NSPERINT Never Secure Period Interrupt Enable

Value	Description
0	No effect.
1	A never secure period failure generates an interrupt in the programmable secure area.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Value	Description
0	No effect.
1	The programmable secure interrupt threshold failure interrupt is enabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Enable

Value	Description
0	No effect.
1	The programmable secure repeat threshold failure interrupt is enabled.

Bit 0 – PERINT Period Interrupt Enable

Value	Description
0	No effect.
1	The programmable secure period failure interrupt is enabled.

23.5.16 DWDT Programmable Secure Watchdog Interrupt Disable Register

Name: PS_WDT_IDR
Offset: 0x1198
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 4 – NSRPTHINT Never Secure Repeat Threshold Interrupt Disable

Value	Description
0	No effect.
1	A never secure repeat threshold failure does not generate an interrupt in the programmable secure area.

Bit 3 – NSPERINT Never Secure Period Interrupt Disable

Value	Description
0	No effect.
1	A never secure period failure does not generate an interrupt in the programmable secure area.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Value	Description
0	No effect.
1	The Programmable secure interrupt threshold failure interrupt is disabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Disable

Value	Description
0	No effect.
1	The Programmable secure repeat threshold failure interrupt is disabled.

Bit 0 – PERINT Period Interrupt Disable

Value	Description
0	No effect.
1	The Programmable secure period failure interrupt is disabled.

23.5.17 DWDT Programmable Secure Watchdog Interrupt Status Register

Name: PS_WDT_ISR
Offset: 0x119C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – NSRPTHINT Never Secure Repeat Threshold Interrupt Status

Value	Description
0	No never secure repeat threshold failure has occurred in the programmable secure watchdog since the last read of PS_WDT_ISR.
1	At least one never secure repeat threshold failure occurred in the programmable secure watchdog since the last read of PS_WDT_ISR.

Bit 3 – NSPERINT Never Secure Period Interrupt Status

Value	Description
0	No never secure period failure has occurred in the programmable secure watchdog since the last read of PS_WDT_ISR.
1	At least one never secure period failure occurred in the programmable secure watchdog since the last read of PS_WDT_ISR.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Status

Value	Description
0	No level threshold failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.
1	At least one level threshold failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.

Bit 1 – RPTHINT Repeat Threshold Interrupt Status

Value	Description
0	No repeat threshold failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.
1	At least one repeat threshold failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.

Bit 0 – PERINT Period Interrupt Status

Value	Description
0	No period failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.
1	At least one period failure has occurred in the Programmable secure watchdog since the last read of PS_WDT_ISR.

23.5.18 DWDT Programmable Secure Watchdog Interrupt Mask Register

Name: PS_WDT_IMR
Offset: 0x11A0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				NSRPTHINT	NSPERINT	LVLINT	RPTHINT	PERINT
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – NSRPTHINT Never Secure Repeat Threshold Interrupt Mask

Value	Description
0	Programmable secure interrupt on never secure RPTHINT interrupt is disabled.
1	Programmable secure interrupt on never secure RPTHINT interrupt is enabled.

Bit 3 – NSPERINT Never Secure Period Interrupt Mask

Value	Description
0	Programmable secure interrupt on never secure PERINT interrupt is disabled.
1	Programmable secure interrupt on never secure PERINT interrupt is enabled.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Value	Description
0	Interrupt on LVLINT is disabled.
1	Interrupt on LVLINT is enabled.

Bit 1 – RPTHINT Repeat Threshold Interrupt Mask

Value	Description
0	Interrupt on RPTHINT is disabled.
1	Interrupt on RPTHINT is enabled.

Bit 0 – PERINT Period Interrupt Mask

Value	Description
0	Interrupt on PERINT is disabled.
1	Interrupt on PERINT is enabled.

23.5.19 DWDT Never Secure Level Limit Register

Name: NS_WDT_LVLLIM
Offset: 0x11A4
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					LVLMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1
Bit	23	22	21	20	19	18	17	16
	LVLMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
					LVLMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	LVLMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – LVLMAX[11:0] Level Maximum
Defines the maximum value that can be applied on NS_WDT_IL.LVLTH.
If the currently defined LVLTH value is higher than the LVLMAX to be configured, LVLMAX value is applied to LVLTH.

Bits 11:0 – LVLMIN[11:0] Level Minimum
Defines the minimum value that can be applied on NS_WDT_IL.LVLTH.
If the currently defined LVLTH value is lower than the LVLMIN to be configured, LVLMIN value is applied to LVLTH.

23.5.20 DWDT Never Secure Repeat Threshold Limit Register

Name: NS_WDT_RLIM
Offset: 0x11A8
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTHMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	RPTHMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
					RPTHMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RPTHMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – RPTHMAX[11:0] Repeat Threshold Maximum
Defines the maximum value that can be applied to NS_WDT_WL.RPTH.
If the currently defined RPTH value is higher than the RPTHMAX to be configured, RPTHMAX value is applied to RPTH.

Bits 11:0 – RPTHMIN[11:0] Repeat Threshold Minimum
Defines the minimum value that can be applied to NS_WDT_WL.RPTH.
If the currently defined RPTH value is lower than the RPTHMIN to be configured, RPTHMIN value is applied to RPTH.

23.5.21 DWDT Never Secure Period Limit Register

Name: NS_WDT_PLIM
Offset: 0x11AC
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					PERMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
					PERMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – PERMAX[11:0] Period Maximum
Defines the maximum value that can be applied to NS_WDT_WL.PERIOD.
If the currently defined PERIOD value of the NS_WDT is higher than the PERMAX to be configured, PERMAX value is applied to PERIOD.

Bits 11:0 – PERMIN[11:0] Period Minimum
Defines the minimum value that can be applied to NS_WDT_WL.PERIOD.
If the currently defined PERIOD value of the NS_WDT is lower than the PERMIN to be configured, PERMIN value is applied to PERIOD.

24. Reset Controller (RSTC)

24.1 Description

The Reset Controller (RSTC) handles all the resets of the system without any external components. It reports which reset occurred last.

The RSTC is driven by Power-on Reset (POR) cells, software, an external reset pin, and peripheral events.

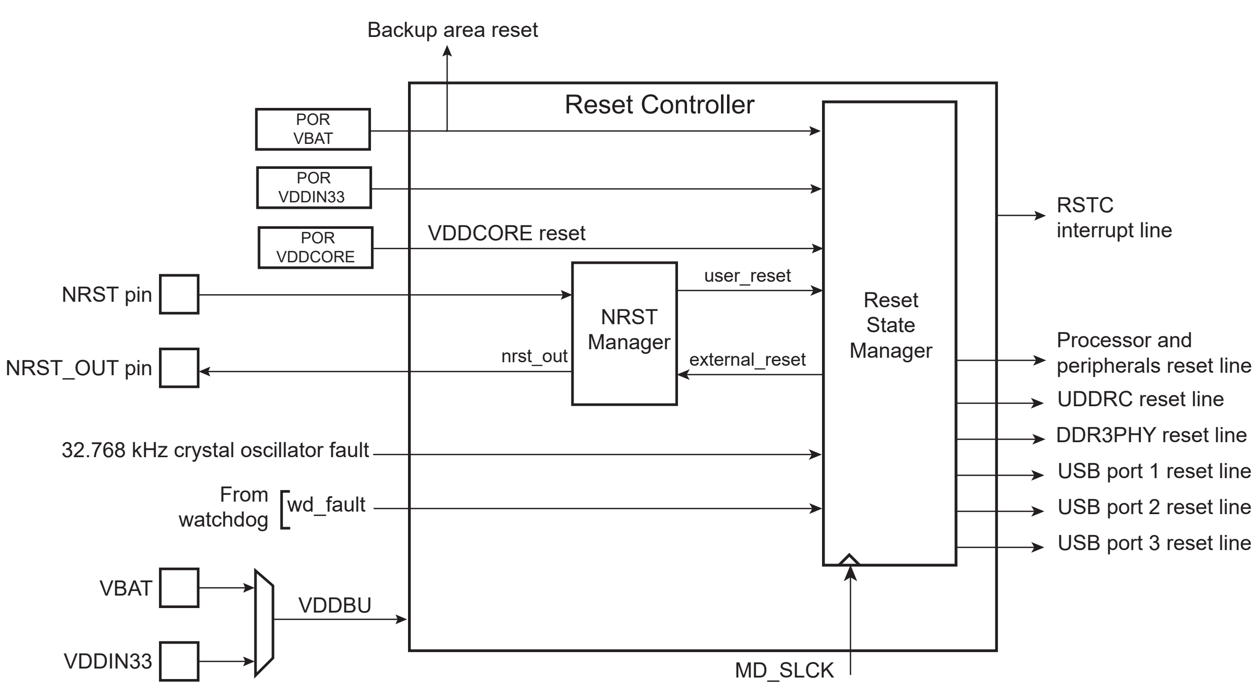
The RSTC drives simultaneously the External reset and the Peripheral and Processor resets.

24.2 Embedded Characteristics

- Driven by Embedded Power-on Reset, Software, External Reset Pin and Peripheral Events
- Management of All System Resets, Including
  - External devices through an I/O multiplexed output reset pin
  - Processor
  - Peripheral set
- Reset Source Status
  - Status of the last reset
  - Either VDDCORE and VBAT POR reset, Software reset, User reset, Watchdog reset, 32.768 kHz Crystal Oscillator, ULP2 Low-power mode exit

24.3 Block Diagram

Figure 24-1. RSTC Block Diagram



24.4 Functional Description

The RSTC is made up of an NRST manager and a reset state manager. The RSTC clock is MD_SLCK (monitoring domain slow clock). The RSTC generates the following reset signals:

- Processor reset line (also resets the Watchdog Timer)
- Entire set of embedded peripherals reset line
- NRST_OUT pin

Note: Processor and peripheral reset lines are driven in the same way.

These internal reset signals are asserted by the RSTC, either on events generated by peripherals, events on NRST pin, or on software action. The reset state manager controls the generation of reset signals and drives the NRST_OUT pin when required.

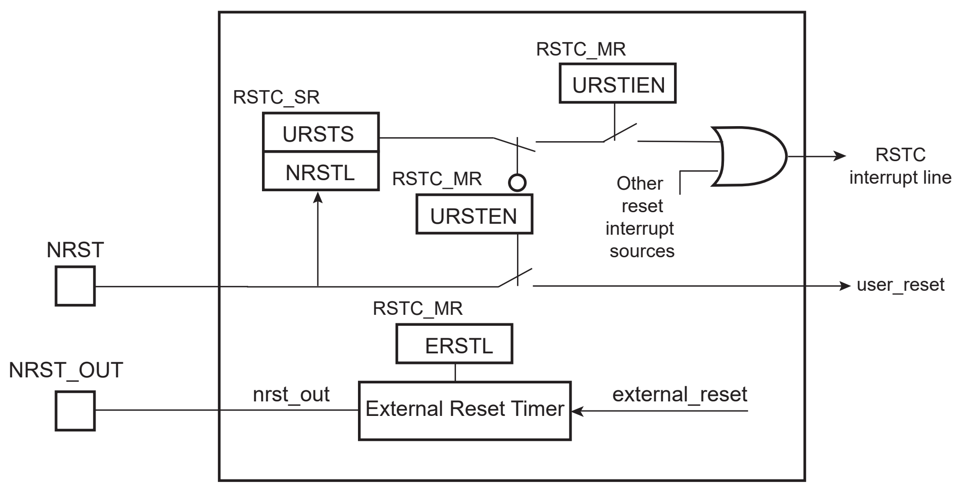
The NRST manager asserts the NRST_OUT pin during a programmable time, thus controlling external device resets.

The Mode register (RSTC_MR), used to configure the RSTC, is powered by VDDBU.

24.4.1 NRST Manager

The NRST manager samples the NRST pin and drives the NRST_OUT pin low when required by the reset state manager. See the following figure.

Figure 24-2. NRST Pin Management



24.4.1.1 NRST Signal or Interrupt

The NRST manager handles the NRST input line asynchronously if RSTC_MR.URSTASYNC =1. When the NRST input is low, a user reset is immediately reported to the Reset State manager and the internal reset signals are asserted even if there is a clock failure on MD_SLCK (safe reset).

The NRST manager handles the NRST input line synchronously if RSTC_MR.URSTASYNC=0. When the line is low, it is first resynchronized on slow clock before it is reported to the Reset State manager. In both cases, when the NRST goes from low to high, the internal reset is synchronized with the monitoring slow clock to provide a safe internal de-assertion of reset (if enabled).

If RSTC_MR.URSTEN=0, the assertion of the NRST input pin does not trigger a VDDCORE domain reset.

The level of the pin NRST is reported in NRSTL of the Status register (RSTC_SR).

As soon as the pin NRST is asserted (low level), RSTC_SR.URSTS=1. This bit is cleared on read.

If RSTC_MR.URSTIEN=1, the assertion of NRST pin triggers an interrupt rather than a VDDCORE reset.

24.4.1.2 NRST_OUT External Reset Control

The RSTC can be configured to assert the external reset line (NRST_OUT). The NRST_OUT pin is driven low for a time programmed by RSTC_MR.ERSTL. This assertion duration lasts $2^{(ERSTL+1)}$ MD_SLCK cycles. This assertion duration time is in the range of 60 µs to 2 seconds. If ERSTL=0, a two slow clock period duration is generated on the NRST_OUT pin.

This feature allows the NRST_OUT line to be compliant with any external devices connected on the system reset (i.e., when external devices require a longer start-up time than the processor system).

24.4.1.3 USB Reset Control

The RSTC controls the resets of the USB Controller through bit USBRSTx of the Generic Reset register (RSTC_GRSTR).

24.4.1.4 DDR Reset Control

The RSTC controls the reset of the DDR Controller through RSTC_GRSTR.DDRRST.

24.4.2 Reset States

The reset state manager handles the different reset sources and generates the internal reset signals. It reports the reset status in RSTC_SR.RSTTYP. RSTC_SR.RSTTYP is updated when the Processor reset is released.

If more than one reset event occurred since the last read of RST_SR, the field RSTTYP reports the first reset that occurred.

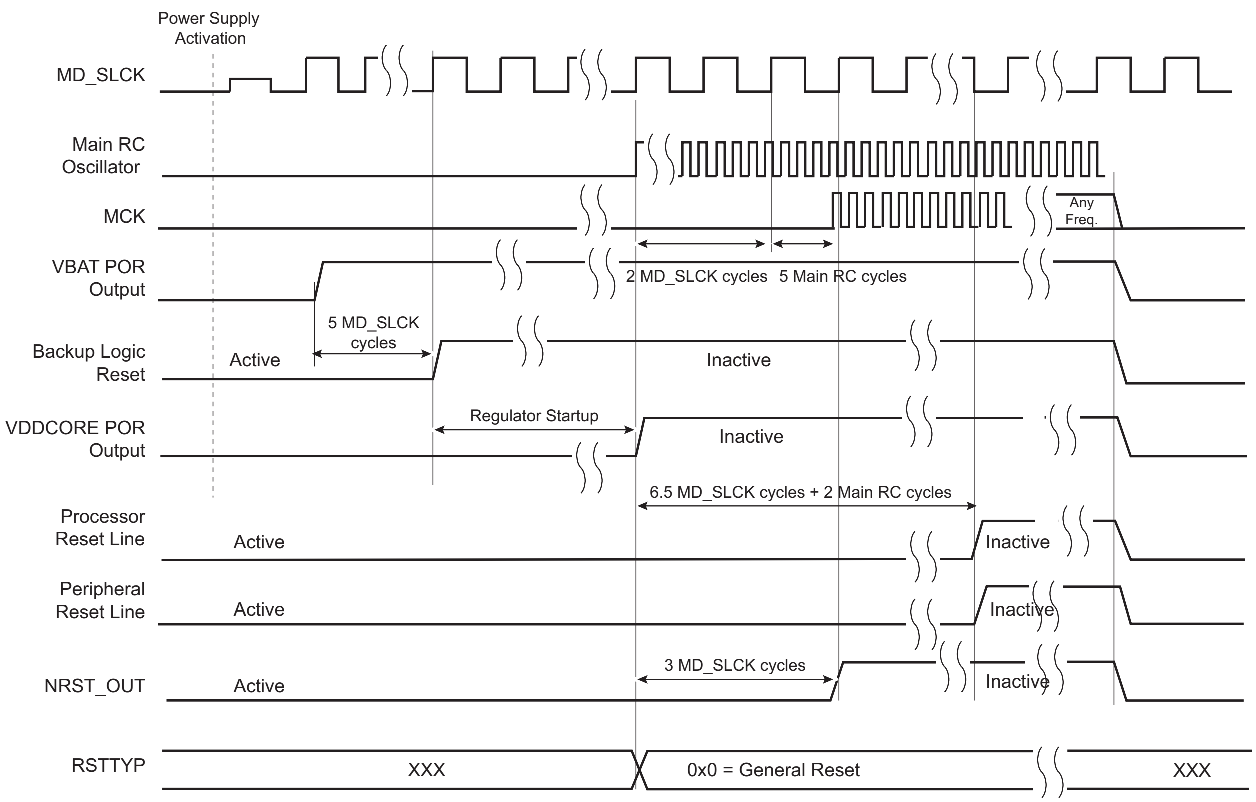
24.4.2.1 General Reset

A general reset occurs when a VBAT Power-on reset is detected. The internal VDDCORE reset signal is asserted when a general reset occurs. All the reset signals are released and RSTC_SR.RSTTYP reports a general reset.

The NRST_OUT line rises two cycles after the VDDCORE reset line, as ERSTL defaults at value 0x0.

The following figure shows how the general reset affects the reset signals.

Figure 24-3. General Reset Timing Diagram (VDDBU supplied by VBAT)



24.4.2.2 Backup Exit Reset

A Backup reset occurs when the chip exits from Backup mode. While exiting Backup mode, the VDDCORE reset signal is de-asserted.

RSTC_SR.RSTTYP is updated to report a Backup reset.

24.4.2.3 32.768 kHz Crystal Oscillator Failure Detection Reset

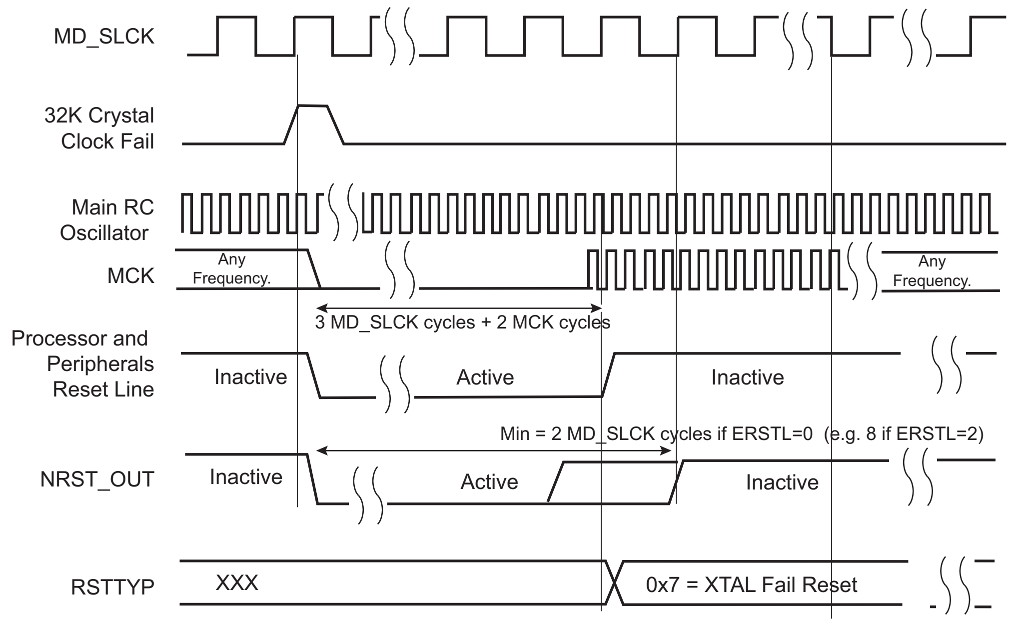
The 32.768 kHz Crystal Oscillator Failure Detection reset is done when the 32.768 kHz crystal oscillator frequency monitoring circuitry in the PMC detects a failure and RSTC_MR.SCKSW is written to '1'. This reset lasts three slow clock cycles.

When RSTC_MR.SCKSW is written to ‘0’, the 32.768 kHz crystal oscillator fault has no impact on the RSTC.

During the 32.768 kHz Crystal Oscillator Failure Detection reset, the Processor reset and the Peripheral reset are asserted. The NRST_OUT line is also asserted, depending on the value of RSTC_MR.ERSTL.

When the 32.768 kHz crystal oscillator failure generates a VDDCORE reset, PMC_SR.XT32KERR is automatically cleared by the Peripheral and Processor resets.

Figure 24-4. 32.768 kHz Crystal Oscillator Failure Detection Reset Timing Diagram

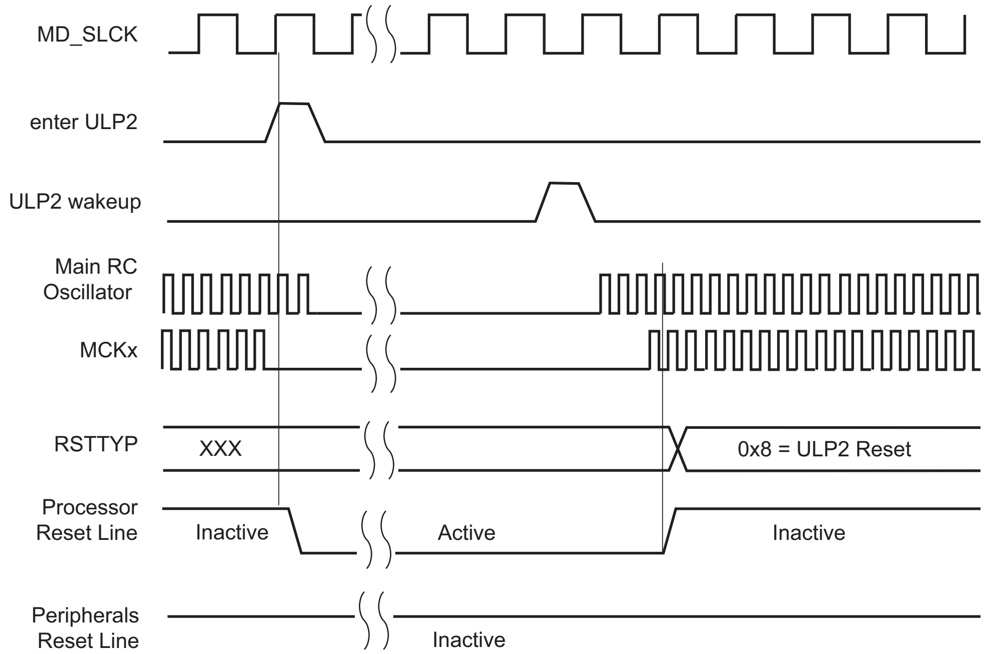


24.4.2.4 ULP Mode 2 Reset

The ULP Mode 2 reset is entered when a WFE event occurs while CKGR_MOR.ULP2 is set to ‘1’ in the PMC.

When this reset occurs, only the Processor reset is asserted. The VDDCPU power supply can be switched off during the ULP mode 2 (refer to “ULP Mode 2” in the section “Power Management Controller (PMC)”.

Figure 24-5. ULP Mode 2 Reset Timing Diagram



24.4.2.5 Watchdog Reset

The Watchdog reset is entered when a watchdog fault occurs. This reset lasts three MD_SLCK cycles.

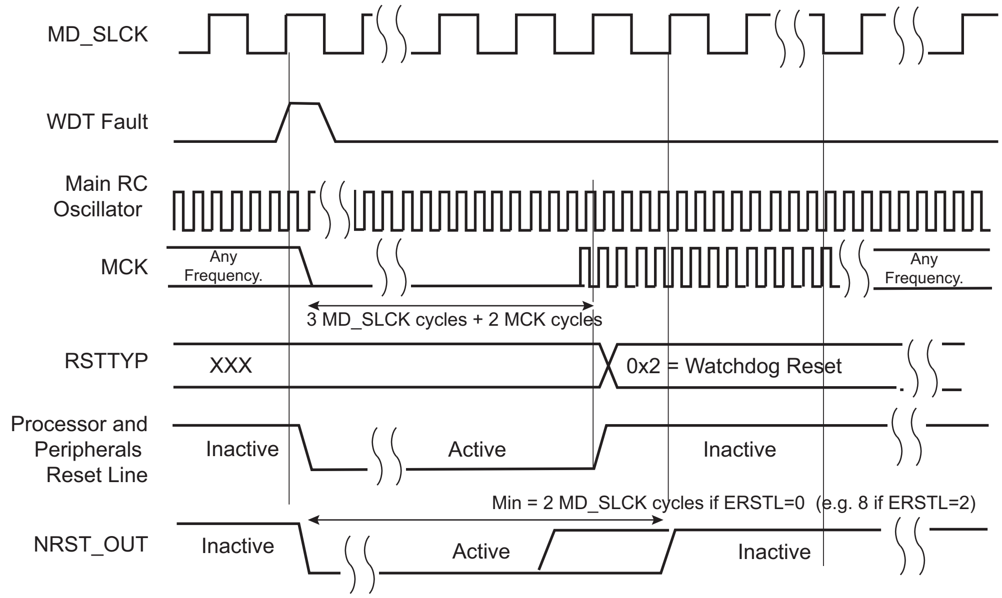
When in Watchdog reset, the Processor reset and the Peripheral reset are asserted. The NRST_OUT line is also asserted, depending on the value of RSTC_MR.ERSTL. However, the resulting low level on NRST_OUT does not result in a User reset state.

The WDT is reset by the Processor reset signal. As the watchdog fault always causes a Processor reset if WDT_MR.WDRSTEN is written to ‘1’, the WDT is always reset after a Watchdog reset, and the Watchdog is enabled by default and with a period set to a maximum.

When WDT_MR.WDRSTEN is written to ‘0’, the watchdog fault has no impact on the RSTC.

After a watchdog overflow occurs, the report on the RSTC_SR.RSTTYP may differ (either WDT_RST or USER_RST) depending on the external components driving the NRST pin. For example, if the NRST line is driven through a resistor and a capacitor (NRST pin debouncer), the reported value is USER_RST if the low-to-high transition is greater than one MD_SLCK cycle.

Figure 24-6. Watchdog Reset Timing Diagram



24.4.2.6 Software Reset

The RSTC offers commands to assert the different reset signals. These commands are performed by writing the Control register (RSTC_CR) with the following bits at ‘1’:

- RSTC_CR.PROCRST: Writing a ‘1’ to PROCRST resets the processor and all the embedded peripherals, including the memory system and, in particular, the Remap Command.
- RSTC_CR.EXTRST: Writing a ‘1’ to EXTRST asserts low the NRST_OUT pin during a time defined by the field RSTC_MR.ERSTL.

The Software reset is entered if at least one of these bits is written to ‘1’ by the software. All these commands can be performed independently or simultaneously. The Software reset lasts three MD_SLCK cycles.

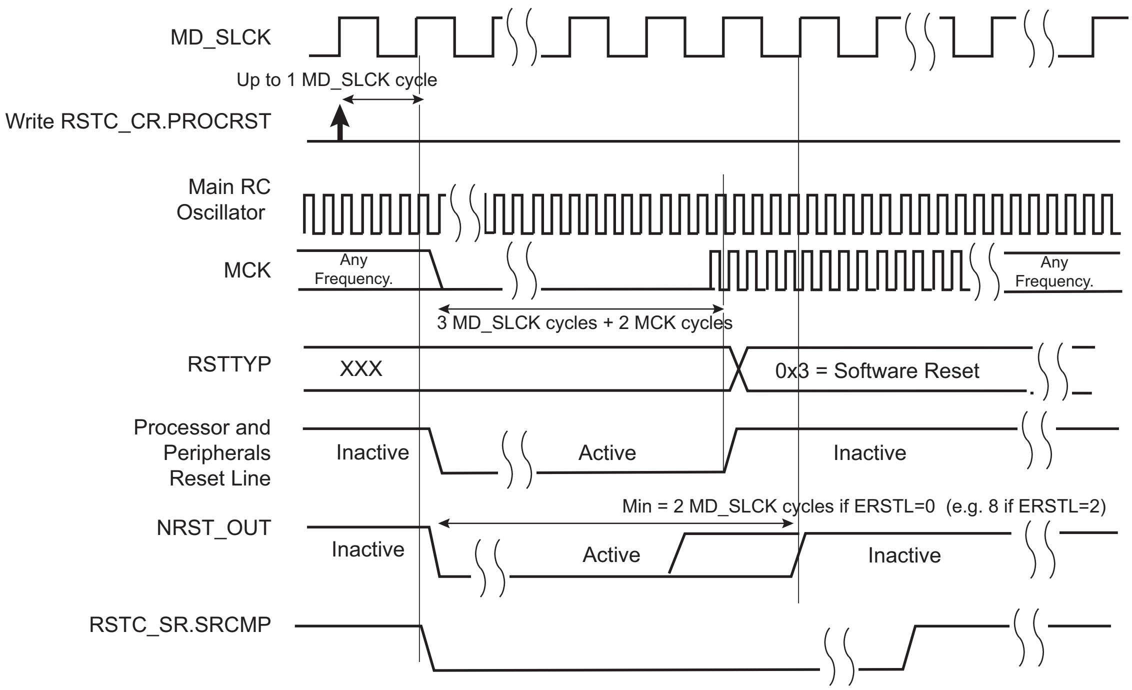
The internal reset signals are asserted as soon as the register write is performed. They are released when the Software reset has ended, i.e., synchronously to MD_SLCK.

If EXTRST is written to ‘1’, the NRST_OUT pin is asserted depending on the configuration of RSTC_MR.ERSTL. However, the resulting falling edge on NRST_OUT does not lead to a User reset.

If and only if the RSTC_CR.PROCRST is written to ‘1’, the RSTC reports the software status in field RSTC_SR.RSTTYP. Other software resets are not reported in RSTTYP.

As soon as a software operation is detected, RSTC_SR.SRCMP is written to ‘1’. SRCMP is cleared at the end of the Software reset. No other Software reset can be performed while SRCMP=‘1’, and writing any value in the RSTC_CR has no effect.

Figure 24-7. Software Reset Timing Diagram (PROCRST)



24.4.2.7 User Reset

The User reset is entered when a low level is detected on the NRST pin and RSTC_MR.URSTEN =1. If URSTASYNC=1, a falling edge of the NRST input signal immediately asserts internal reset lines. If URSTASYNC=0, the NRST input signal is resynchronized and internal reset lines are asserted once a falling edge has been detected on the resynchronized NRST input signal.

The Processor reset and the Peripheral reset are asserted.

The User reset is released when NRST rises, after a two-cycle resynchronization time and a 2-cycle processor start-up. The processor clock is re-enabled as soon as NRST is confirmed high.

When the Processor reset signal is released, RSTC_SR.RSTTYP is loaded with the value 0x4, indicating a User reset.

The NRST manager ensures that the NRST_OUT line is asserted as programmed in the field ERSTL. However, if NRST is driven low externally, the internal reset lines remain asserted until NRST rises.

Figure 24-8. User Reset State (URSTASYNC = '0')

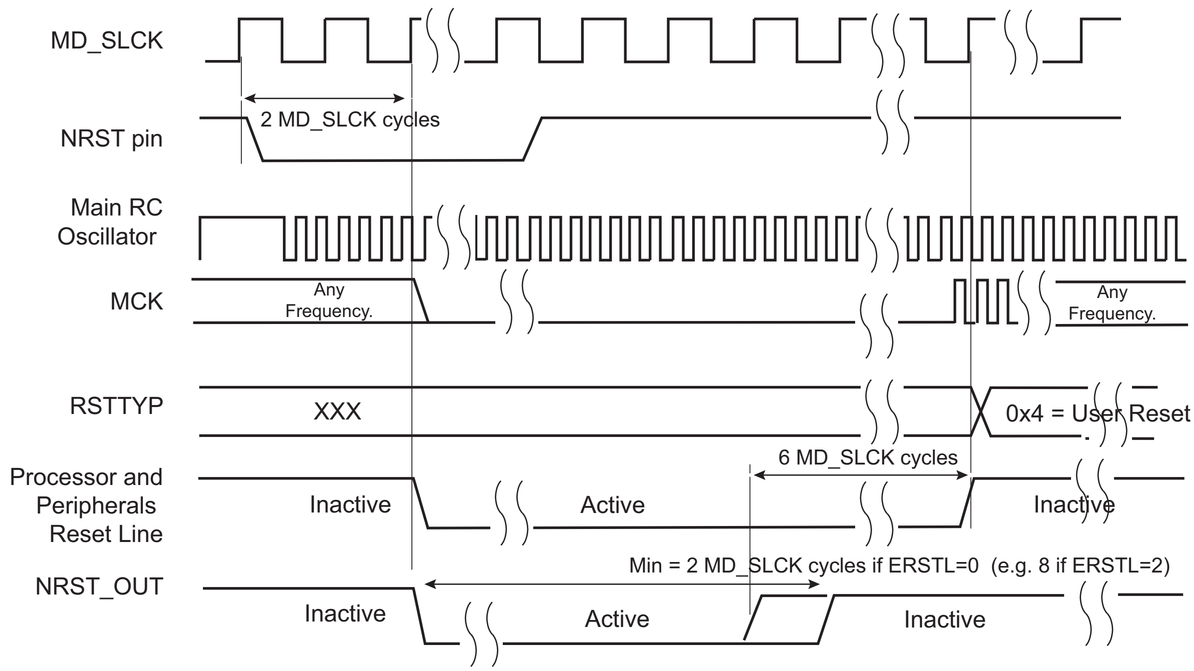
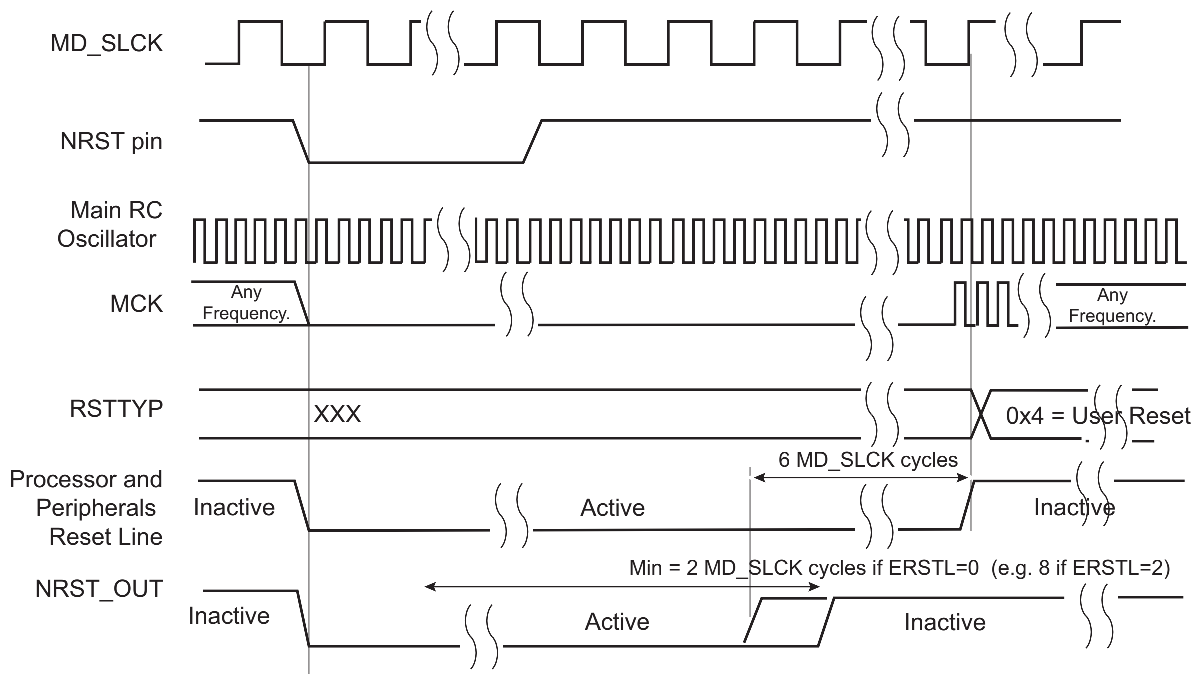


Figure 24-9. User Reset State (URSTASYNC = '1')



24.4.3 Reset State Priorities

The reset state manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. Backup reset
3. 32.768 kHz Crystal Failure Detection reset
4. Watchdog reset

5. Software reset
6. User reset

Specific cases are listed below:

- When in User reset:
  – A watchdog event is impossible because the WDT is being reset by the Processor reset signal.
  – A Software reset is impossible, since the Processor reset is being activated.
- When in Software reset:
  – A watchdog event has priority over the current state.
  – The NRST has no effect.
- When in Watchdog reset:
  – The Processor reset is active and so a Software reset cannot be programmed.
  – A User reset cannot be entered.

24.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RSTC_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0					EXTRST			PROCRST
0x04	RSTC_SR	31:24								
		23:16							SRCMP	NRSTL
		15:8					RSTTYP[3:0]			
		7:0								URSTS
0x08	RSTC_MR	31:24	KEY[7:0]							
		23:16								
		15:8					ERSTL[3:0]			
		7:0				URSTIEN		URSTASYNC	SCKSW	URSTEN
0x0C ... 0xE3	Reserved									
0xE4	RSTC_GRSTR	31:24								
		23:16								
		15:8								
		7:0		USB_RST3	USB_RST2	USB_RST1		DDR_PHY_RST		DDR_RST

24.5.1 RSTC Control Register

Name: RSTC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					EXTRST			PROCRST
Access					W			W
Reset					–			–

Bits 31:24 – KEY[7:0] System Reset Key

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 3 – EXTRST External Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, asserts the NRST_OUT pin.

Bit 0 – PROCRST Processor Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, resets the processor and all the embedded peripherals.

24.5.2 RSTC Status Register

Name: RSTC_SR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

The reset value assumes that a general reset has been performed, subject to change if other types of reset are generated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SRCMP	NRSTL
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
					RSTTYP[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
								URSTS
Access								R
Reset								0

Bit 17 – SRCMP Software Reset Command in Progress
When set, this bit indicates that a software reset command is in progress and that no further software reset should be performed until the end of the current one. This bit is automatically cleared at the end of the current software reset.

Value	Description
0	No software command is being performed by the RSTC. The RSTC is ready for a software command.
1	A software reset command is being performed by the RSTC. The RSTC is busy.

Bit 16 – NRSTL NRST Pin Level
Registers the NRST pin level sampled on each MCK rising edge.

Bits 11:8 – RSTTYP[3:0] Reset Type
This field reports the cause of the last processor reset. Reading RSTC_SR does not reset this field.

Value	Name	Description
0	GENERAL_RST	First power-up reset
1	BACKUP_RST	Return from Backup mode
2	WDT_RST	Watchdog fault occurred
3	SOFT_RST	Processor reset required by the software
4	USER_RST	NRST pin detected low
5	–	Reserved
6	–	Reserved
7	SLCK_XTAL_RST	32.768 kHz crystal failure detection fault occurred
8	ULP2_RST	ULP Mode 2 reset

Bit 0 – URSTS User Reset Status
A high-to-low transition of the NRST pin sets URSTS. This transition is also detected on the MCK rising edge. If the user reset is disabled (RSTC_MR.URSTEN = 0) and if the interrupt is enabled by RSTC_MR.URSTIEN, URSTS triggers an interrupt. Reading RSTC_SR resets URSTS and clears the interrupt.

Value	Description
0	No high-to-low edge on NRST happened since the last read of RSTC_SR.
1	At least one high-to-low transition of NRST has been detected since the last read of RSTC_SR.

24.5.3 RSTC Mode Register

Name: RSTC_MR
Offset: 0x08
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ERSTL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				URSTIEN		URSTASYNC	SCKSW	URSTEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	1

Bits 31:24 – KEY[7:0] Write Access Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bits 11:8 – ERSTL[3:0] External Reset Length
This field defines the external reset length. The external reset pin RST_OUT is asserted during a time of $2^{(ERSTL+1)}$ MD_SLCK cycles. This allows assertion duration to be programmed between 60 µs and 2 seconds. Note that synchronization cycles must also be considered when calculating the actual reset length as previously described.

Bit 4 – URSTIEN User Reset Interrupt Enable

Value	Description
0	RSTC_SR.USRTS at ‘1’ has no effect on the RSTC interrupt line.
1	RSTC_SR.USRTS at ‘1’ asserts the RSTC interrupt line if URSTEN = 0.

Bit 2 – URSTASYNC User Reset Asynchronous Control
See NRST Signal or Interrupt for important information on the use of URSTASYNC.

Value	Description
0	The NRST input signal is managed synchronously.
1	The NRST input signal is managed asynchronously. **Note:** This mode cannot be selected if the external bus interface drives an SDR/DDR memory device and another memory on the same bus.

Bit 1 – SCKSW Slow Clock Switching

Value	Description
0	The detection of a 32.768 kHz crystal failure has no effect.
1	The detection of a 32.768 kHz crystal failure resets the logic supplied by VDDCORE.

Bit 0 – URSTEN User Reset Enable

Value	Description
0	The detection of a low level on the NRST pin does not generate a user reset.
1	The detection of a low level on the NRST pin triggers a user reset.

24.5.4 RSTC Generic Reset Register

Name: RSTC_GRSTR
Offset: 0xE4
Reset: 0x00000070
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		USB_RST3	USB_RST2	USB_RST1		DDR_PHY_RST		DDR_RST
Access		R/W	R/W	R/W		R/W		R/W
Reset		1	1	1		0		0

Bits 4, 5, 6 – USB_RSTx USB Reset
0 (POR_DISABLED): POR is disabled.
1 (POR_ENABLED): POR is enabled.

Bit 2 – DDR_PHY_RST DDR PHY Reset

Value	Description
0	DDR PHY reset is asserted.
1	DDR PHY reset is de-asserted.

Bit 0 – DDR_RST DDR Reset

Value	Description
0	DDR controller reset is asserted.
1	DDR controller reset is de-asserted.

25. Real-time Timer (RTT)

25.1 Description

The Real-time Timer (RTT) is built around a 32-bit counter used to count roll-over events of the programmable 16-bit prescaler driven from the 32-kHz slow clock source. It generates a periodic interrupt and/or triggers an alarm on a programmed value.

The RTT can also be configured to be driven by the 1Hz RTC signal, thus taking advantage of a calibrated 1Hz clock.

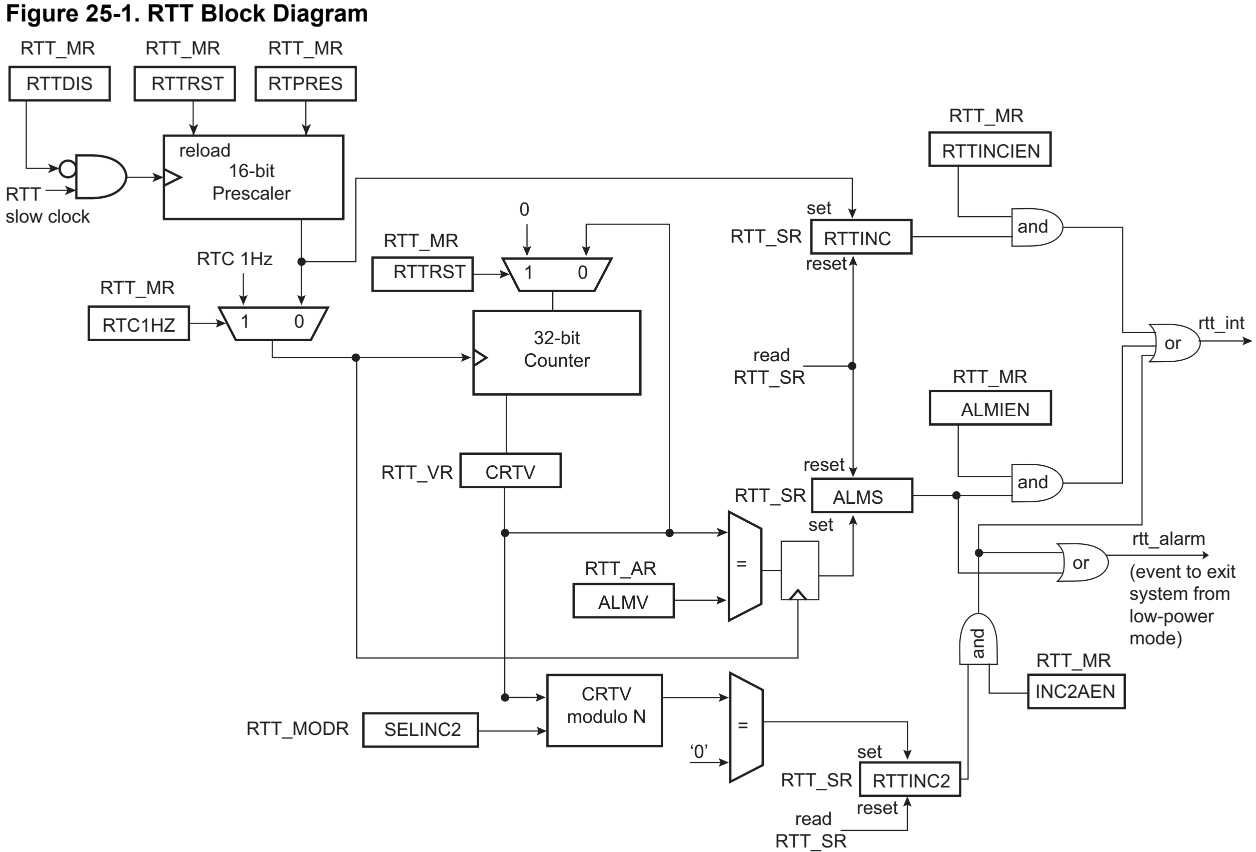
The slow clock source can be fully disabled to reduce power consumption when only an elapsed seconds count is required.

25.2 Embedded Characteristics

- 32-bit Free-running Counter on Prescaled Slow Clock or RTC Calibrated 1Hz Clock
- 16-bit Configurable Prescaler
- Interrupt on Alarm or Counter Increment
- Programmable Event

25.3 Block Diagram

Figure 25-1. RTT Block Diagram



25.4 Functional Description

The programmable 16-bit prescaler value can be configured through the RTPRES field in the RTT Mode register (RTT_MR).

Configuring RTPRES to 0x8000 (default value) corresponds to feeding the real-time counter with a 1Hz signal (if the slow clock is 32.768 kHz). The 32-bit counter can count up to 2^{32} seconds, corresponding to more than 136 years, then roll over to 0. Bit RTTINC in the RTT Status register (RTT_SR) is set each time there is a prescaler roll-over (see the figure below).

The real-time 32-bit counter can also be supplied by the 1Hz RTC clock. This mode is applicable when the RTC 1Hz is calibrated (RTC_MR.CORRECTION ≠ 0) in order to ensure the synchronism between RTC and RTT counters.

Setting RTT_MR.RTC1HZ drives the 32-bit RTT counter from the 1Hz RTC clock. In this mode, RTPRES has no effect on the 32-bit counter.

The prescaler roll-over generates an increment of the RTT counter if RTC1HZ = 0. Otherwise, if RTC1HZ = 1, the RTT counter is incremented every second. RTTINC is set independently from the 32-bit counter increment.

The RTT can also be used as a free-running timer with a lower time-base. The best accuracy is achieved by writing RTT_MR.RTPRES to 3.

Programming RTPRES to 1 or 2 is forbidden.

The CRTV field can be read at any time in the RTT Value register (RTT_VR). As this value can be updated asynchronously with the peripheral bus clock, CRTV must be read twice at the same value to read a correct value.

The current value of the counter is compared with the value written in the RTT Alarm register (RTT_AR). If the counter value matches the alarm, the ALMS bit in the RTT_SR is set. The RTT_AR is set to its maximum value (0xFFFFFFFF) after a reset.

ALMS is always a source of the RTT alarm signal that may be used to exit the system from low power modes (see the RTT Block Diagram above).

The alarm interrupt must be disabled (RTT_MR.ALMIEN must be cleared) when writing a new value to RTT_AR.ALMV.

RTT_SR.RTTINC can be used to start a periodic interrupt. The period is one second when RTT_MR.RTPRES = 0x8000 and the slow clock = 32.768 kHz.

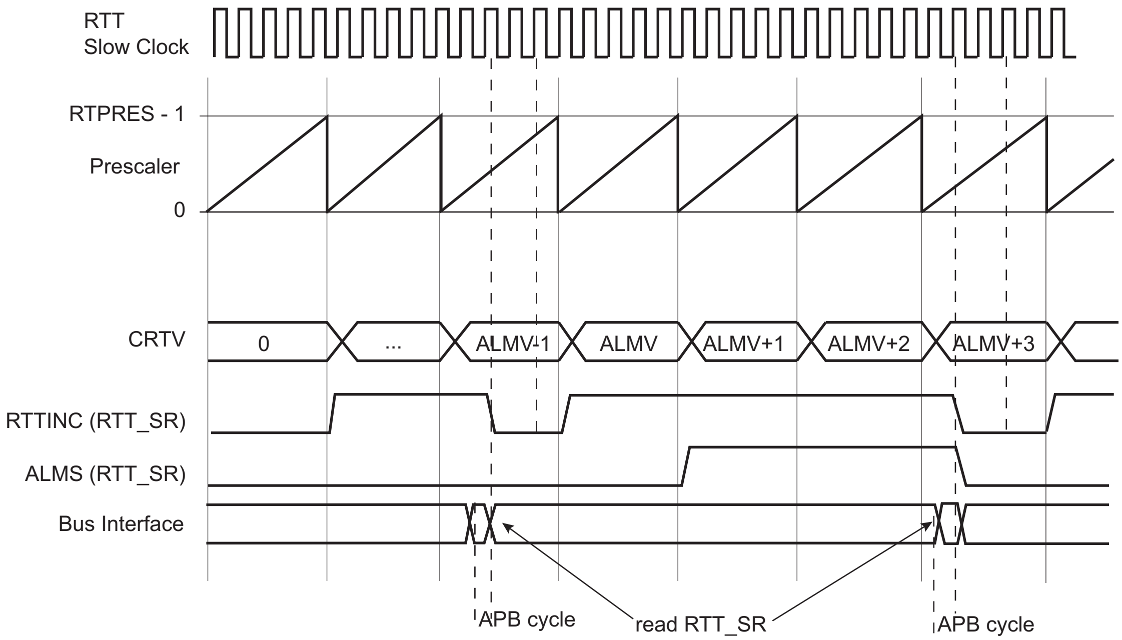
RTT_MR.RTTINCIEN must be cleared prior to writing a new value in RTT_MR. RTPRES.

Reading RTT_SR automatically clears RTT_SR.RTTINC and RTT_SR.ALMS.

Writing RTT_MR.RTTRST immediately reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

When not used, the RTT can be disabled in order to suppress dynamic power consumption in this module. This can be achieved by setting RTT_MR.RTTDIS.

Figure 25-2. RTT Counting



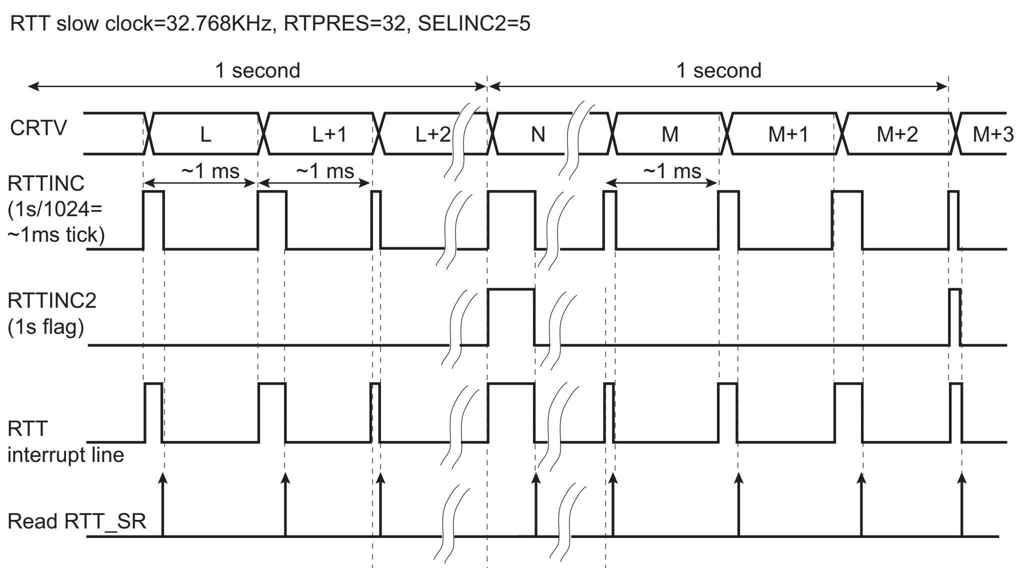
The RTTINC2 flag is set when the number of prescaler roll-overs programmed through the SELINC2 field in the RTT Modulo Selection register (RTT_MODR) has been reached since the last read of RTT_SR.

For example, it is possible to generate 2 sources of interrupt of different periods with flags RTTINC and RTTINC2. If the RTT slow clock frequency is 32.768 kHz and RTPRES=32, the RTTINC flag rises 1024 times per second (less than 1 ms period). If the field SELINC2=5, the RTTINC2 flag rises once per second.

If RTTINC is defined as the unique source of interrupt (RTTINCEN=1, ALMIEN=0 and INC2AEN=0 in RTT_MR), the value read in RTT_SR by the interrupt handler determines if the current interrupt event corresponds to a 1-second event (RTT_SR[2:1]=3) or to a 1-millisecond event (RTT_SR[2:1]=1). See the figure below.

If the bit INC2AEN=1, RTTINC2 flag is also a source for the RTT alarm signal. See RTT Block Diagram.

Figure 25-3. RTTINC2 Behavior



25.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RTT_MR	31:24								RTC1HZ
		23:16			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
		15:8	RTPRES[15:8]							
		7:0	RTPRES[7:0]							
0x04	RTT_AR	31:24	ALMV[31:24]							
		23:16	ALMV[23:16]							
		15:8	ALMV[15:8]							
		7:0	ALMV[7:0]							
0x08	RTT_VR	31:24	CRTV[31:24]							
		23:16	CRTV[23:16]							
		15:8	CRTV[15:8]							
		7:0	CRTV[7:0]							
0x0C	RTT_SR	31:24								
		23:16								
		15:8								
		7:0						RTTINC2	RTTINC	ALMS
0x10	RTT_MODR	31:24								
		23:16								
		15:8								
		7:0						SELINC2[2:0]		
0x14	RTT_TSR	31:24								
		23:16	TSTAMP[23:16]							
		15:8	TSTAMP[15:8]							
		7:0	TSTAMP[7:0]							

25.5.1 Real-time Timer Mode Register

Name: RTT_MR
Offset: 0x00
Reset: 0x00008000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
								RTC1HZ
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	RTPRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RTPRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – RTC1HZ Real-Time Clock 1Hz Clock Selection

Value	Description
0	The RTT 32-bit counter is driven by the 16-bit prescaler roll-over events.
1	The RTT 32-bit counter is driven by the 1Hz RTC clock.

Bit 21 – INC2AEN RTTINC2 Alarm and Interrupt Enable

Value	Description
0	The RTTINC2 flag is not a source of the RTT alarm signal nor a source of interrupt.
1	The RTTINC2 flag is a source of the RTT alarm signal and a source of interrupt.

Bit 20 – RTTDIS Real-time Timer Disable

Value	Description
0	The RTT is enabled.
1	The RTT is disabled (no dynamic power consumption).

Bit 18 – RTTRST Real-time Timer Restart

Value	Description
0	No effect.
1	Reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

Bit 17 – RTTINCIEN Real-time Timer Increment Interrupt Enable

Value	Description
0	The bit RTTINC in RTT_SR has no effect on interrupt.
1	The bit RTTINC in RTT_SR asserts interrupt.

Bit 16 – ALMIEN Alarm Interrupt Enable

Value	Description
0	The bit ALMS in RTT_SR has no effect on interrupt.
1	The bit ALMS in RTT_SR asserts interrupt.

Bits 15:0 – RTPRES[15:0] Real-time Timer Prescaler Value
Defines the number of RTT slow clock periods required to increment the Real-time timer. The RTTINCIEN bit must be cleared prior to writing a new RTPRES value.
RTPRES is defined as follows:

- RTPRES = 0: The prescaler period is equal to 2^{16} * slow clock periods.
- RTPRES = 1 or 2: forbidden.
- RTPRES ≠ 0, 1 or 2: The prescaler period is equal to RTPRES * slow clock periods.

25.5.2 Real-time Timer Alarm Register

Name: RTT_AR
Offset: 0x04
Reset: 0xFFFFFFFF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	ALMV[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ALMV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ALMV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	ALMV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – ALMV[31:0] Alarm Value
When the CRTV value in RTT_VR equals the ALMV field, the ALMS flag is set in RTT_SR.
The alarm interrupt must be disabled (ALMIEN must be cleared in RTT_MR) when writing a new ALMV value.

25.5.3 Real-time Timer Value Register

Name: RTT_VR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CRTV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRTV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRTV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRTV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRTV[31:0] Current Real-time Value
Returns the current value of the RTT.
As CRTV can be updated asynchronously, it must be read twice at the same value.

25.5.4 Real-time Timer Status Register

Name: RTT_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RTTINC2	RTTINC	ALMS
Access						R	R	R
Reset						0	0	0

Bit 2 – RTTINC2 Predefined Number of Prescaler Roll-overs Status (cleared on read)

Value	Description
0	SELINC2 = 0 or the number of prescaler roll-overs programmed through the SELINC2 field in RTT_MODR has not been reached since the last read of the RTT_SR.
1	The number of prescaler roll-overs programmed through the SELINC2 field has been reached since the last read of the RTT_SR.

Bit 1 – RTTINC Prescaler Roll-over Status (cleared on read)

Value	Description
0	No prescaler roll-over occurred since the last read of the RTT_SR.
1	Prescaler roll-over occurred since the last read of the RTT_SR.

Bit 0 – ALMS Real-time Alarm Status (cleared on read)

Value	Description
0	The Real-time Alarm has not occurred since the last read of RTT_SR.
1	The Real-time Alarm occurred since the last read of RTT_SR.

25.5.5 Real-time Timer Modulo Selection Register

Name: RTT_MODR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SELINC2[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SELINC2[2:0] Selection of the 32-bit Counter Modulo to generate RTTINC2 Flag

Value	Name	Description
0	NO_RTTINC2	The RTTINC2 flag never rises.
1	MOD64	The RTTINC2 flag is set when CRTV modulo 64 equals 0.
2	MOD128	The RTTINC2 flag is set when CRTV modulo 128 equals 0.
3	MOD256	The RTTINC2 flag is set when CRTV modulo 256 equals 0.
4	MOD512	The RTTINC2 flag is set when CRTV modulo 512 equals 0.
5	MOD1024	The RTTINC2 flag is set when CRTV modulo 1024 equals 0. Example: If RTPRES=32 then RTTINC2 flag rises once per second if the slow clock is 32.768 kHz.
6	MOD2048	The RTTINC2 flag is set when CRTV modulo 2048 equals 0.
7	MOD4096	The RTTINC2 flag is set when CRTV modulo 4096 equals 0.

25.5.6 Real-time Timer Timestamp Register

Name: RTT_TSR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	TSTAMP[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TSTAMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TSTAMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – TSTAMP[23:0] Real-time Timer Value Timestamp
Each time an event triggers the flag RTT_SR.RTTINC2, RTT_VR.CRTV is copied into RTT_TSR.TSTAMP. The field TSTAMP remains stable until next event RTT_SR.RTTINC2 and can be used for event timestamping.

26. Real-time Clock (RTC)

26.1 Description

The Real-time Clock (RTC) peripheral is designed for very low power consumption. For optimal functionality, the RTC requires an accurate external 32.768 kHz clock, which can be provided by a crystal oscillator.

It combines a complete time-of-day clock with alarm and a Gregorian or Persian calendar, complemented by a programmable periodic interrupt. The alarm and calendar registers are accessed by a 32-bit data bus.

The RTC can also be configured for the UTC time format.

The time and calendar values are coded in binary-coded decimal (BCD) format. The time format can be 24-hour mode or 12-hour mode with an AM/PM indicator.

Updating time and calendar fields and configuring the alarm fields are performed by a parallel capture on the 32-bit data bus. An entry control is performed to avoid loading registers with incompatible BCD format data or with an incompatible date according to the current month/year/century.

A clock divider calibration circuitry can be used to compensate for crystal oscillator frequency variations.

An RTC output can be programmed to generate several waveforms, including a prescaled clock derived from 32.768 kHz.

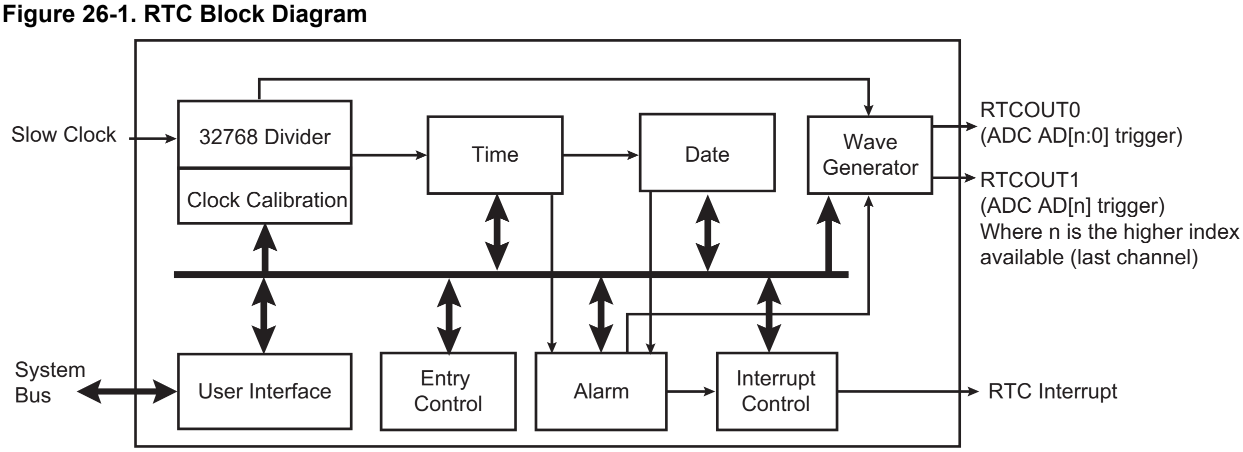
Timestamping capability reports the first and last occurrences of tamper events.

26.2 Embedded Characteristics

- Full Asynchronous Design for Ultra Low-Power Consumption
- Gregorian, UTC and Persian Modes Supported
- Programmable Periodic Interrupt
- Safety/Security Features:
  - Valid time and date programming check
  - On-the-fly time and date validity check
- Counters Calibration Circuitry to Compensate for Crystal Oscillator Variations
- Waveform Generation for ADC Trigger Event
- Tamper Timestamping Registers
- Register Write Protection

26.3 Block Diagram

Figure 26-1. RTC Block Diagram



26.4 Product Dependencies

26.4.1 Power Management

The Real-time Clock is continuously clocked at 32.768 kHz. The Power Management Controller (PMC) has no effect on RTC behavior.

26.4.2 Interrupt

Within the System Controller, the RTC interrupt is OR-wired with all the other module interrupts.

Only one System Controller interrupt line is connected on one of the internal sources of the interrupt controller.

RTC interrupt requires the interrupt controller to be programmed first.

When a System Controller interrupt occurs, the service routine must first determine the cause of the interrupt. This is done by reading each status register of the System Controller peripherals successively.

26.5 Functional Description

The RTC provides a full binary-coded decimal (BCD) clock that includes century (19/20), year (with leap years), month, date, day, hours, minutes and seconds reported in RTC Time Register (RTC_TIMR) and RTC Calendar Register (RTC_CALR).

The RTC can operate in UTC mode, giving the number of seconds elapsed since a reference time defined by the user (the UTC standard—ISO 8601—reference time is the 30th of June 1972). In this mode, the timefield is 32 bits wide and coded in hexadecimal format.

The valid year range is up to 2099 in Gregorian mode (or 1300 to 1499 in Persian mode).

The RTC can operate in 24-hour mode or in 12-hour mode with an AM/PM indicator.

Corrections for leap years are included (all years divisible by 4 being leap years except 1900). This is correct up to the year 2099.

The RTC can generate configurable waveforms on RTCOUT0/1 outputs.

26.5.1 Reference Clock

The reference clock is the slow clock. It can be driven externally by a 32.768 kHz crystal, or internally.

26.5.2 Timing

In Gregorian and Persian modes, the RTC is updated in real time at one-second intervals in Normal mode for the counters of seconds, at one-minute intervals for the counter of minutes and so on.

In UTC mode, the RTC is updated in real-time at one-second intervals (32-bit UTC counter default configuration).

Due to the asynchronous operation of the RTC with respect to the rest of the chip, to ensure that the value read in the RTC registers (century, year, month, date, day, hours, minutes, seconds) are valid and stable, it is necessary to read these registers twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

26.5.3 Alarm

In Gregorian and Persian modes, the RTC has five programmable fields: month, date, hours, minutes and seconds.

Each of these fields can be enabled or disabled to match the alarm condition:

- If all the fields are enabled, an alarm flag is generated (the corresponding flag is asserted and an interrupt generated if enabled) at a given month, date, hour/minute/second.
- If only the “seconds” field is enabled, then an alarm is generated every minute.

Depending on the fields that are enabled in the RTC Calendar Alarm register (RTC_CALALR) and the RTC Time Alarm register(RTC_TIMALR), a large number of possibilities are available to the user ranging from minutes to 365/366 days.

Note: To change one of the RTC_TIMALR.SEC, MIN, HOUR and/or RTC_CALALR.DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_TIMALR or RTC_CALALR. The first access clears the enable corresponding to the field to change (RTC_TIMALR.SECEN, MINEN, HOUREN and/or RTC_CALALR.DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (RTC_TIMALR.SEC, MIN, HOUR and/or RTC_CALALR.DATE, MONTH). The third access is required to re-enable the field by writing 1 in RTC_TIMALR.SECEN, MINEN, HOUREN and/or RTC_CALALR.DATEEN, MTHEN.

In UTC mode, RTC_TIMALR must be configured to set the UTC alarm value and bit 0 in RTC_CALALR must be used to enable or disable the UTC alarm. If the UTC alarm is enabled, the alarm is generated once the UTC time matches the programmed UTC_TIME alarm field.

To change the UTC_TIME alarm field, proceed as follows:

1. Disable the UTC alarm by clearing RTC_CALALR.UTCEN if it is not already cleared.
2. Change the RTC_TIMALR.UTC_TIME alarm value.
3. Re-enable the UTC alarm by setting RTC_CALALR.UTCEN.

26.5.4 Error Checking when Programming

Verification on user interface data is performed when accessing the century, year, month, date, day, hours, minutes, seconds and alarms. A check is performed on illegal BCD entries such as illegal date of the month with regard to the year and century configured.

If one of the time fields is not correct, the data is not loaded into the register/counter and a flag is set in the validity register. The user can not reset this flag. It is reset as soon as an acceptable value is programmed. This avoids any further side effects in the hardware. The same procedure is followed for the alarm.

The following checks are performed:

1. Century (check if it is in range 19–20 or 13–14 in Persian mode)
2. Year (BCD entry check)
3. Date (check range 01–31)
4. Month (check if it is in BCD range 01–12, check validity regarding “date”)
5. Day (check range 1–7)
6. Hour (BCD checks: in 24-hour mode, check range 00–23 and check that AM/PM flag is not set if RTC is set in 24-hour mode; in 12-hour mode check range 01–12)
7. Minute (check BCD and range 00–59)
8. Second (check BCD and range 00–59)

Notes:

1. If the 12-hour mode is selected by means of the Mode register (RTC_MR), a 12-hour value can be programmed and the returned value on RTC_TIMR will be the corresponding 24-hour value. The entry control checks the value of the AM/PM indicator (bit 22 of RTC_TIMR) to determine the range to be checked.
2. In UTC mode, no check is performed on the entries. The RTC does not report any failure.

26.5.5 RTC Internal Free-Running Counter Error Checking

To improve the reliability and security of the RTC, a permanent check is performed on the internal free-running counters to report non-BCD or invalid date/time values.

An error is reported by RTC_SR.TDERR if an incorrect value has been detected. The flag can be cleared by setting the RTC_SCCR.TDERRCLR.

In all cases, RTC_SR.TDERR is set again if the source of the error has not been cleared before clearing RTC_SR.TDERR. The clearing of the source of such error can be done by reprogramming a correct value on RTC_CALR and/or RTC_TIMR.

The RTC internal free-running counters may automatically clear the source of RTC_SR.TDERR due to their roll-over (i.e., every 10 seconds for SECONDS[3:0] in RTC_TIMR). In this case, RTC_SR.TDERR is held high until a clear command is asserted by writing a 1 in RTC_SCCR.TDERRCLR.

26.5.6 Updating Time/Calendar

26.5.6.1 Gregorian and Persian Modes

To update time and date, the RTC must be stopped by setting the corresponding field in the Control register (RTC_CR). RTC_CR.UPDTIM must be set to update time fields (hour, minute, second) and RTC_CR.UPDCAL must be set to update calendar fields (century, year, month, date, day).

RTC_SR.ACKUPD must then be read to 1 by either polling RTC_SR or by enabling the acknowledge update interrupt by writing RTC_IER.ACKUPD to '1'. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to the Time register (RTC_TIMR), the Calendar register (RTC_CALR), or both.

Once the update is finished, the user must write a '0' in RTC_CR.UPDTIM and/or RTC_CR.UPDCAL.

The timing sequence of the time/calendar update is described in the figure below.

When entering the Programming mode of the calendar fields, the time fields remain enabled. When entering the Programming mode of the time fields, both the time and the calendar fields are stopped. This is due to the location of the calendar logical circuitry (downstream for low-power considerations).

In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing these bits to '1'.

Figure 26-2. Time/Calendar Update Timing Diagram

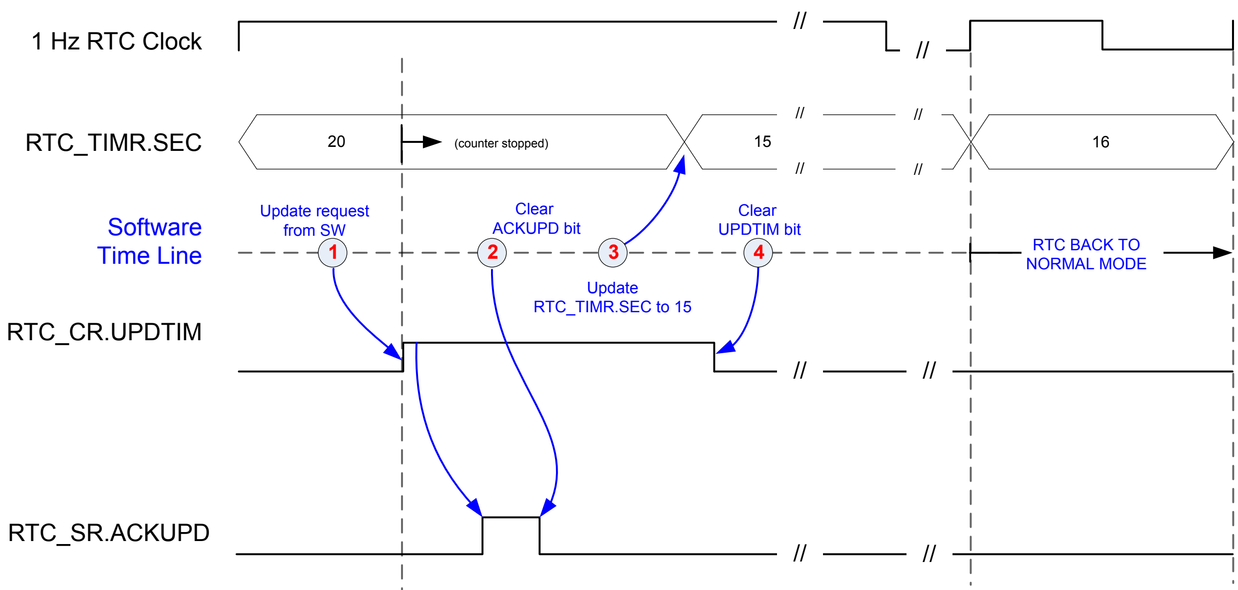
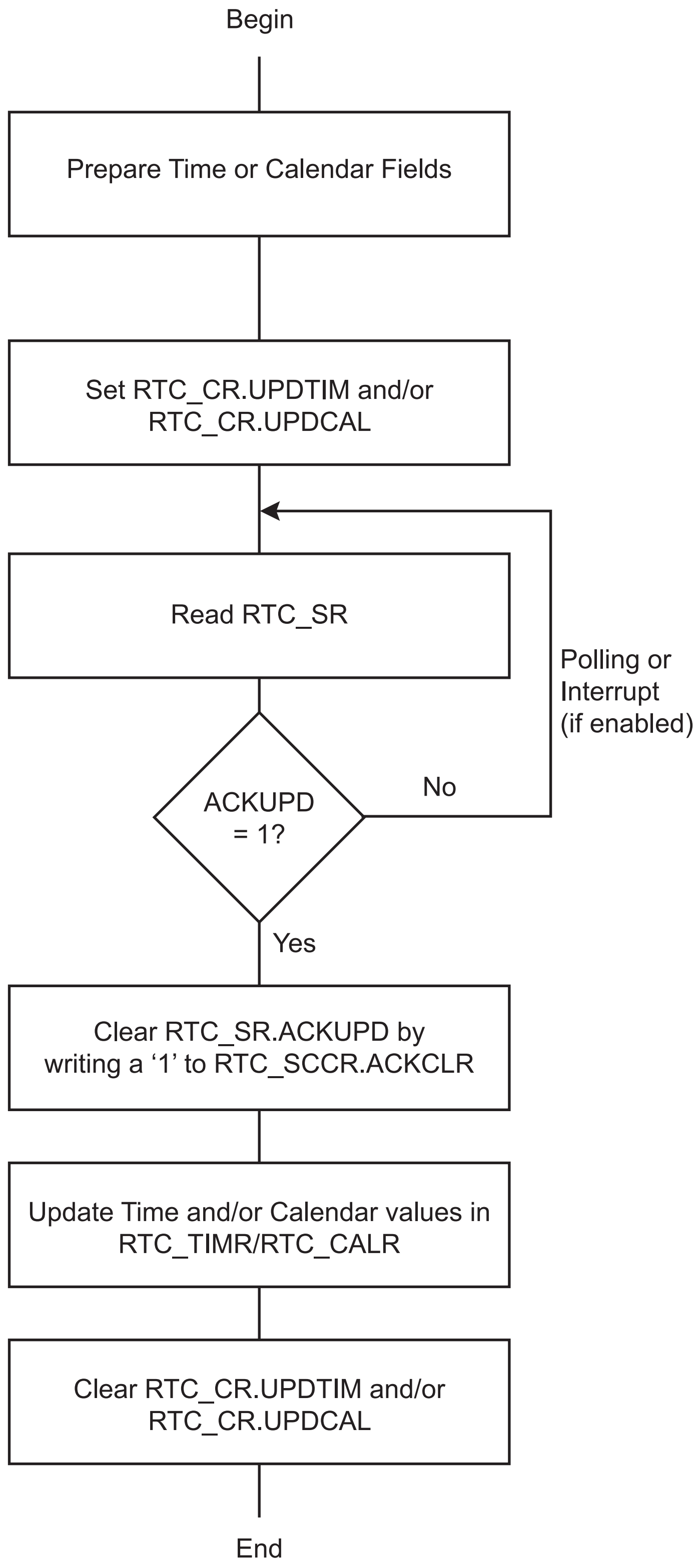


Figure 26-3. Gregorian and Persian Modes Update Sequence



26.5.6.2 UTC Mode

To update the UTC time, the RTC must be stopped by writing a 1 in RTC_CR.UPDTIM and RTC_CR.UPDCAL.

RTC_SR.ACKUPD must then be read to 1 by either polling RTC_SR or by enabling the acknowledge update interrupt by writing a 1 in RTC_IER.ACKUP. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to RTC_TIMR.

Once the update is finished, the user must write a 0 in RTC_CR.UPDTIM and a 0 in RTC_CR.UPCAL.

In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing a 1 in these bits.

The timing sequence of the UTC time update is described in the figure below.

Figure 26-4. UTC Time Update Timing Diagram

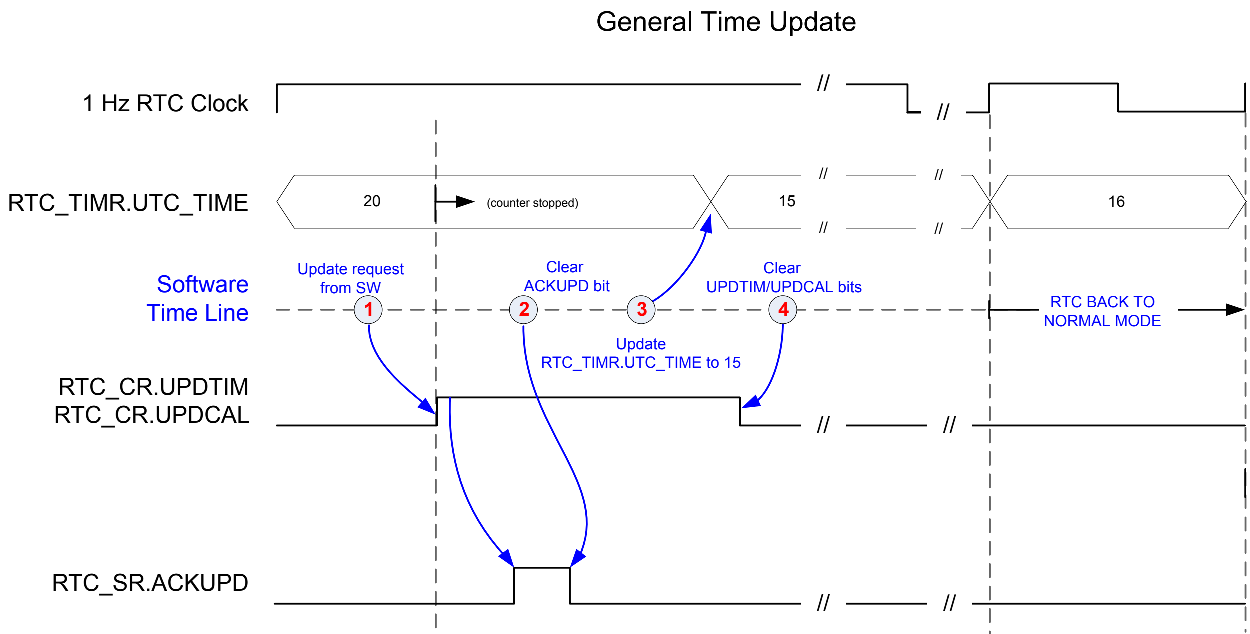
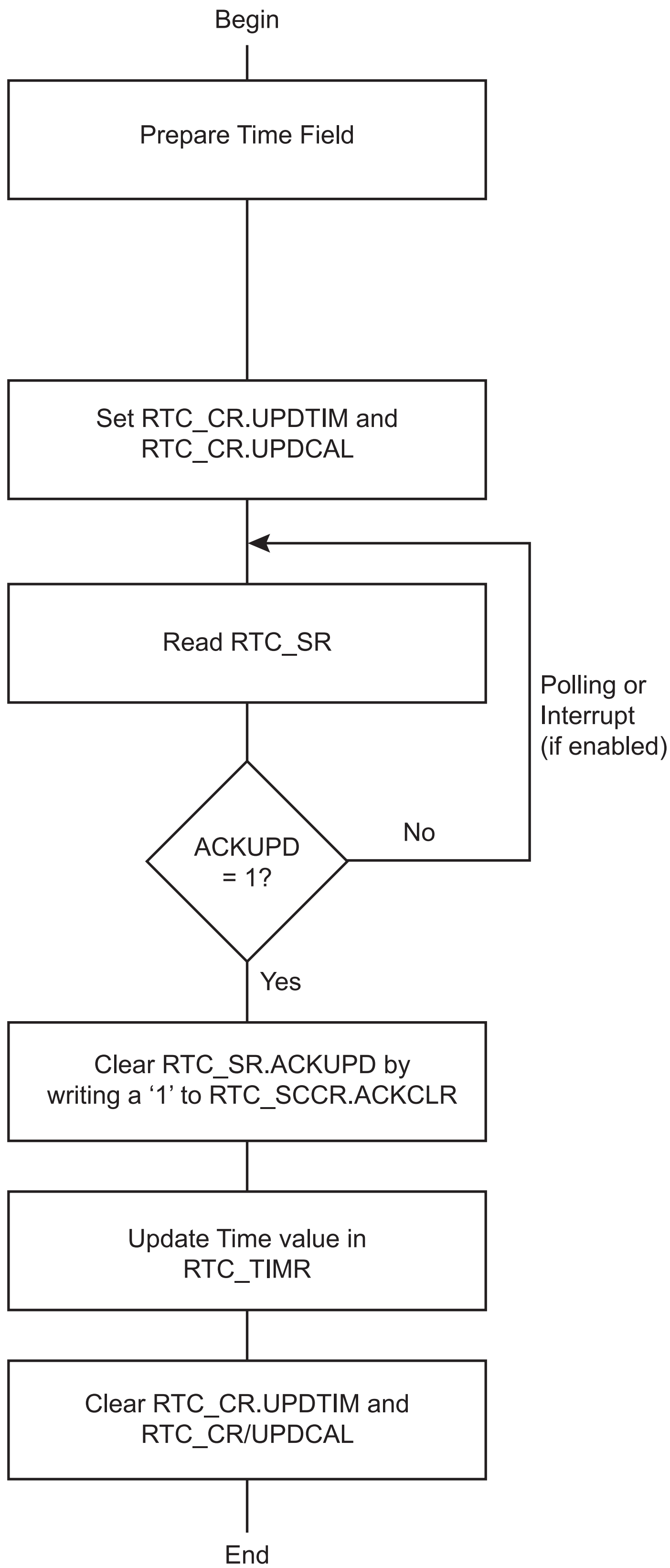


Figure 26-5. UTC Mode Update Sequence



26.5.7 RTC Accurate Clock Calibration

The crystal oscillator that drives the RTC may not be as accurate as expected mainly due to temperature variation. The RTC is equipped with circuitry able to correct slow clock crystal drift.

To compensate for possible temperature variations over time, this accurate clock calibration circuitry can be programmed on-the-fly and also programmed during application manufacturing, in order to correct the crystal frequency accuracy at room temperature (20–25°C). The typical clock drift range at room temperature is ±20 ppm.

In the device operating temperature range, the 32.768 kHz crystal oscillator clock inaccuracy can be up to -200 ppm.

The RTC clock calibration circuitry allows positive or negative correction in a range of 1.5 ppm to 1950 ppm.

The calibration circuitry is fully digital. Thus, the configured correction is independent of temperature, voltage, process, etc., and no additional measurement is required to check that the correction is effective.

If the correction value configured in the calibration circuitry results from an accurate crystal frequency measure, the remaining accuracy is bounded by the values listed below:

- Below 1 ppm, for an initial crystal drift between 1.5 ppm up to 20 ppm, and from 30 ppm to 90 ppm
- Below 2 ppm, for an initial crystal drift between 20 ppm up to 30 ppm, and from 90 ppm to 130 ppm
- Below 5 ppm, for an initial crystal drift between 130 ppm up to 200 ppm

The calibration circuitry does not modify the 32.768 kHz crystal oscillator clock frequency but it acts by slightly modifying the 1 Hz clock period from time to time. The correction event occurs every 1 + [(20 - (19 x HIGHPPM)) x CORRECTION] seconds. When the period is modified, depending on the sign of the correction, the 1 Hz clock period increases or reduces by around 4 ms. Depending on the CORRECTION, NEGPPM and HIGHPPM values configured in RTC_MR, the period interval between two correction events differs.

Figure 26-6. Calibration Circuitry

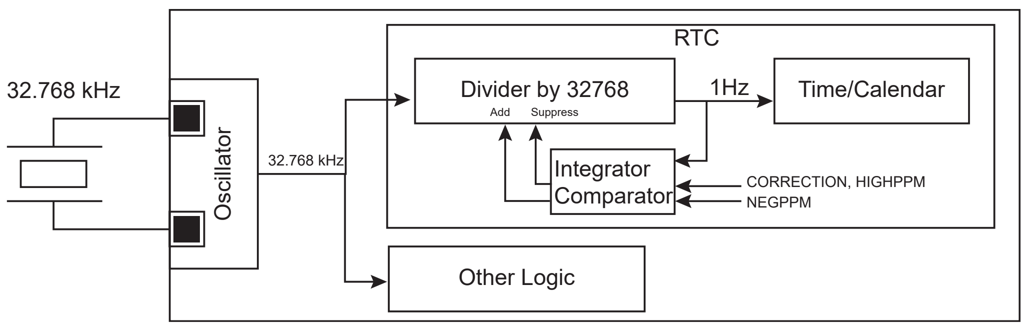
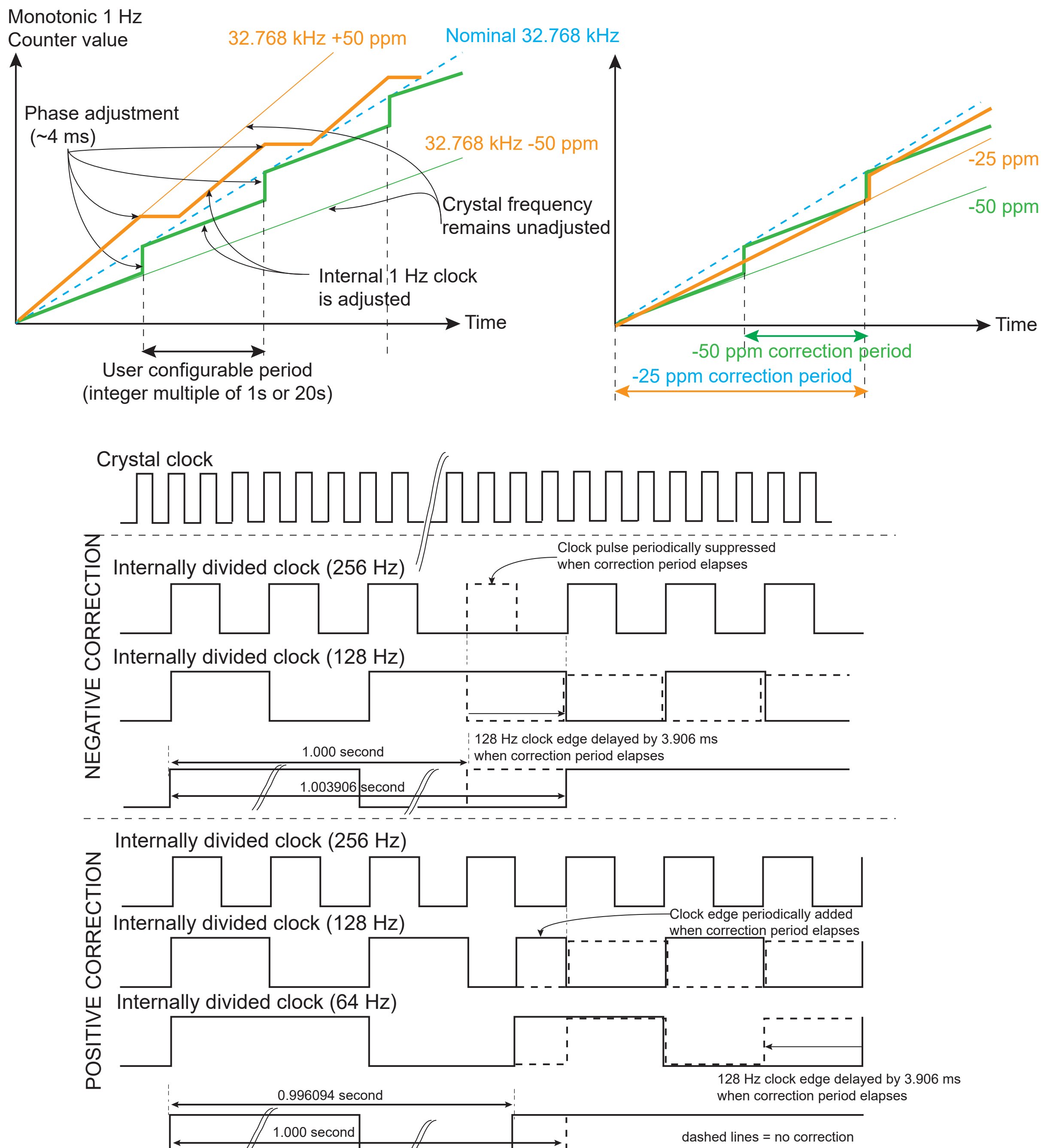


Figure 26-7. Calibration Circuitry Waveforms



The inaccuracy of a crystal oscillator at typical room temperature (±20 ppm at 20–25 °C) can be compensated if a reference clock/signal is used to measure such inaccuracy. This kind of calibration operation can be set up during the final product manufacturing by means of measurement equipment embedding such a reference clock. The correction of value must be programmed into RTC_MR, and this value is kept as long as the circuitry is powered (backup area). Removing the backup power supply cancels this calibration. This room temperature calibration can be further processed by means of the networking capability of the target application.

To ease the comparison of the inherent crystal accuracy with the reference clock/signal during manufacturing, an internal prescaled 32.768 kHz clock derivative signal can be assigned to drive RTC output. To accommodate the measure, several clock frequencies can be selected among 1 Hz, 32 Hz, 64 Hz, 512 Hz.

The clock calibration correction drives the internal RTC counters but can also be observed in the RTC output when one of the following three frequencies 1 Hz, 32 Hz or 64 Hz is configured. The correction is not visible in the RTC output if 512 Hz frequency is configured.

Note that this adjustment does not take into account the temperature variation.

The frequency drift (up to -200 ppm) due to temperature variation can be compensated using a reference time if the application can access such a reference. If a reference time cannot be used, a temperature sensor can be placed close to the crystal oscillator in order to get the operating temperature of the crystal oscillator. Once obtained, the temperature may be converted using a lookup table (describing the accuracy/temperature curve of the crystal oscillator used) and RTC_MR configured accordingly. The calibration can be performed on-the-fly. This adjustment method is not based on a measurement of the crystal frequency/drift and therefore can be improved by means of the networking capability of the target application.

If no crystal frequency adjustment has been done during manufacturing, it is still possible to make adjustments. In the case where a reference time of the day can be obtained through a LAN/WAN network, it is possible to calculate the drift of the application crystal oscillator by comparing the values read on RTC_TIMR and RTC_CALR and programming RTC_MR.HIGHPPM and RTC_MR.CORRECTION according to the difference measured between the reference time and those of RTC_TIMR and RTC_CALR.

26.5.8 Waveform Generation

Waveforms can be generated in order to take advantage of the RTC inherent prescalers while the RTC is the only powered circuitry (Low-power mode of operation, Backup mode) or in any active mode. Entering Backup or Low-power operating modes does not affect the waveform generation outputs.

The outputs RTCOUT0 and RTCOUT1 can be configured to provide several types of waveforms. The figure below illustrates the different signals available to generate RTCOUT0 and RTCOUT1.

The RTC waveforms are internally routed to ADC trigger events. These events can be configured to provide several types of waveforms. The figure below illustrates the different signals available to generate the waveforms. Two different triggers can be generated at a time. The first is configured in RTC_MR.OUT0 while the second is configurable in RTC_MR.OUT1. OUT0 manages the trigger for channel AD[n:0] (where n is the higher index available (last channel)), while OUT1 manages the channel AD[n] only for specific modes. See the section "Analog to Digital Converter (ADC)" for selection of the measurement triggers and associated modes of operation.

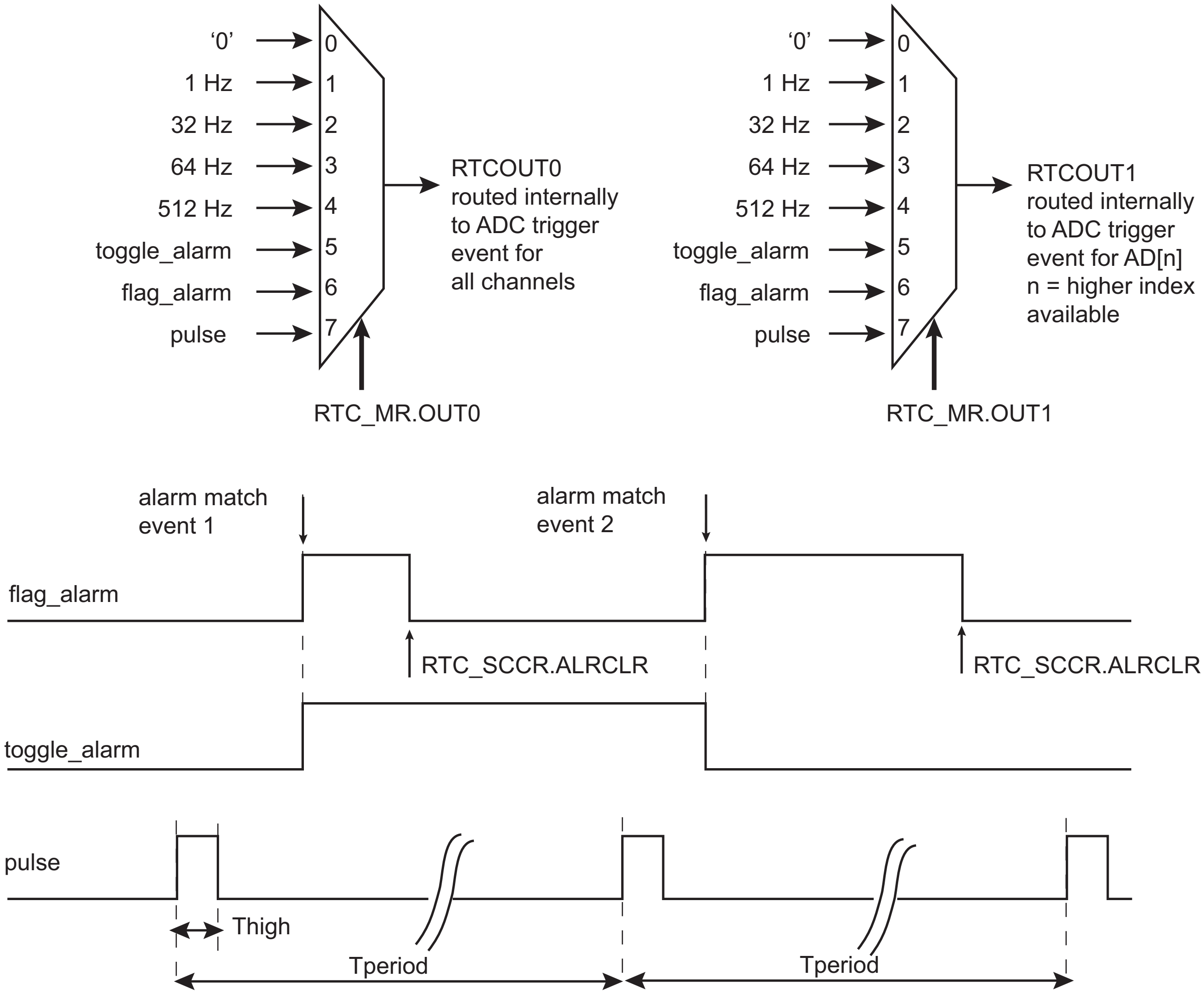
The first selection choice sticks the associated output at 0. (This is the reset value and it can be used at any time to disable the waveform generation).

Selection choices 1 to 4 respectively select 1 Hz, 32 Hz, 64 Hz and 512 Hz.

Selection choice 5 provides a toggling signal when the RTC alarm is reached.

Selection choice 6 provides a copy of the alarm flag, so the associated output is set high (logical 1) when an alarm occurs and immediately cleared when software clears the alarm interrupt source.

Figure 26-8. Waveform Generation



26.5.9 Tamper Timestamping

As soon as a tamper is detected, the tamper counter is incremented and the RTC stores the time of the day, the date and the source of the tamper event in registers located in the backup area. Up to two tamper events can be stored.

In UTC mode, only the UTC time is stored. The date information is not relevant.

The tamper counter saturates at 15. Once this limit is reached, the exact number of tamper occurrences since the last read of stamping registers cannot be known.

The first set of timestamping registers (RTC_TSTR0, RTC_TSDR0, RTC_TSSR0) cannot be overwritten. Once they have been written, all data are stored until the registers are reset. Thus these registers store the first tamper occurrence after a read.

The second set of timestamping registers (RTC_TSTR1, RTC_TSDR1, RTC_TSSR1) are overwritten each time a tamper event is detected. Thus the date and the time data of the first and the second stamping registers may be equal. This occurs when the tamper counter value carried on RTC_TSTR0.TEVCNT equals 1. Thus this second set of registers stores the last occurrence of tamper before a read.

Reading a set of timestamping registers requires three accesses, one for the time of the day, one for the date and one for the tamper source.

Reading the third part (RTC_TSSR0/1) of a timestamping register set clears the whole content of the registers (time, date and tamper source) and makes the timestamping registers available to store a new event.

26.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RTC_CR	31:24								
		23:16							CALEVSEL[1:0]	
		15:8							TIMEVSEL[1:0]	
		7:0							UPDCAL	UPDTIM
0x04	RTC_MR	31:24			TPERIOD[1:0]			THIGH[2:0]		
		23:16		OUT1[2:0]				OUT0[2:0]		
		15:8	HIGHPPM	CORRECTION[6:0]						
		7:0				NEGPPM		UTC	PERSIAN	HRMOD
0x08	RTC_TIMR	31:24								
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x08	RTC_TIMR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x0C	RTC_CALR	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0x10	RTC_TIMALR	31:24								
		23:16	HOUREN	AMPM	HOUR[5:0]					
		15:8	MINEN	MIN[6:0]						
		7:0	SECEN	SEC[6:0]						
0x10	RTC_TIMALR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x14	RTC_CALALR	31:24	DATEEN		DATE[5:0]					
		23:16	MTHEN			MONTH[4:0]				
		15:8								
		7:0								
0x14	RTC_CALALR (UTC_MODE)	31:24								
		23:16								
		15:8								
		7:0								UTCEN
0x18	RTC_SR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
0x1C	RTC_SCCR	31:24								
		23:16								
		15:8								
		7:0			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
0x20	RTC_IER	31:24								
		23:16								
		15:8								
		7:0			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
0x24	RTC_IDR	31:24								
		23:16								
		15:8								
		7:0			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
0x28	RTC_IMR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CAL	TIM	SEC	ALR	ACK

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	RTC_VER	31:24								
		23:16								
		15:8								
		7:0					NVCALALR	NVTIMALR	NVCAL	NVTIM
0x30 ... 0xAF	Reserved									
0xB0	RTC_TSTR0	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0xB0	RTC_TSTR0 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0xB4	RTC_TSDR0	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0xB4	RTC_TSDR0 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0xB8	RTC_TSSR0	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8								
		7:0					JTAG	TST		DWDTSW
0xBC	RTC_TSTR1	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0xBC	RTC_TSTR1 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0xC0	RTC_TSDR1	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0xC0	RTC_TSDR1 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0xC4	RTC_TSSR1	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8								
		7:0					JTAG	TST		DWDTSW

26.6.1 RTC Control Register

Name: RTC_CR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							CALEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							TIMEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							UPDCAL	UPDTIM
Access							R/W	R/W
Reset							0	0

Bits 17:16 – CALEVSEL[1:0] Calendar Event Selection
The event that generates the flag CALEV in RTC_SR depends on the value of CALEVSEL
In UTC mode, this field has no effect on RTC_SR.

Value	Name	Description
0	WEEK	Week change (every Monday at time 00:00:00)
1	MONTH	Month change (every 01 of each month at time 00:00:00)
2	YEAR	Year change (every January 1 at time 00:00:00)
3	YEAR	Reserved

Bits 9:8 – TIMEVSEL[1:0] Time Event Selection
The event that generates the flag TIMEV in RTC_SR depends on the value of TIMEVSEL.
In UTC mode, this field has no effect on RTC_SR.

Value	Name	Description
0	MINUTE	Minute change
1	HOUR	Hour change
2	MIDNIGHT	Every day at midnight
3	NOON	Every day at noon

Bit 1 – UPDCAL Update Request Calendar Register
Calendar counting consists of day, date, month, year and century counters. Calendar counters can be programmed once this bit is set and acknowledged by the RTC_SR.ACKUPD bit.
In UTC mode, both UPDTIM and UPDCAL must be set to '1' in order to update the UTC time value.

Value	Name	Description
0	CLOSE_UPDATE	No effect or, if UPDCAL has been previously written to 1, stops the update procedure.
1	START_UPDATE	Stops the RTC calendar counting.

Bit 0 – UPDTIM Update Request Time Register

Time counting consists of second, minute and hour counters. Time counters can be programmed once this bit is set and acknowledged by the RTC_SR.ACKUPD bit.

In UTC mode, both UPDTIM and UPDCAL must be set to '1' in order to update the UTC time value.

Value	Name	Description
0	CLOSE_UPDATE	No effect or, if UPDTIM has been previously written to 1, stops the update procedure.
1	STARTE_UPDATE	Stops the RTC time counting.

26.6.2 RTC Mode Register

Name: RTC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			TPERIOD[1:0]			THIGH[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		OUT1[2:0]				OUT0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	HIGHPPM	CORRECTION[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				NEGPPM		UTC	PERSIAN	HRMOD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 29:28 – TPERIOD[1:0] Period of the Output Pulse

Value	Name	Description
0	P_1S	1 second
1	P_500MS	500 ms
2	P_250MS	250 ms
3	P_125MS	125 ms

Bits 26:24 – THIGH[2:0] High Duration of the Output Pulse

Value	Name	Description
0	H_31MS	31.2 ms
1	H_16MS	15.6 ms
2	H_4MS	3.91 ms
3	H_976US	976 μs
4	H_488US	488 μs
5	H_122US	122 μs
6	H_30US	30.5 μs
7	H_15US	15.2 μs

Bits 22:20 – OUT1[2:0] RTCOUT1 Output and ADC Last Channel Trigger Event Source Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at ‘0’
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises

Value	Name	Description
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bits 18:16 – OUT0[2:0] RTCOUT0 Output and All ADC Channel Trigger Event Source Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at ‘0’
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bit 15 – HIGHPPM HIGH PPM Correction
If the absolute value of the correction to be applied is lower than 30 ppm, it is recommended to clear HIGHPPM. HIGHPPM set to 1 is recommended for 30 ppm correction and above.
Formula:
If HIGHPPM = 0, then the clock frequency correction range is from 1.5 ppm up to 98 ppm. The RTC accuracy is less than 1 ppm for a range correction from 1.5 ppm up to 30 ppm.
The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\mathrm{CORRECTION} = \frac{3906}{20 \times \mathrm{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to being programmed into CORRECTION field.
If HIGHPPM = 1, then the clock frequency correction range is from 30.5 ppm up to 1950 ppm. The RTC accuracy is less than 1 ppm for a range correction from 30.5 ppm up to 90 ppm.
The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\mathrm{CORRECTION} = \frac{3906}{\mathrm{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to be programmed into CORRECTION field.
If NEGPPM is set to 1, the ppm correction is negative (used to correct crystals that are faster than the nominal 32.768 kHz).

Value	Name	Description
0	DISABLED	Lower range ppm correction with accurate correction (below 30ppm correction).
1	ENABLED	Higher range ppm correction with accurate correction (higher than 30ppm correction).

Bits 14:8 – CORRECTION[6:0] Slow Clock Correction

Value	Name	Description
0	DISABLED	No correction
1–127	–	The slow clock will be corrected according to the formula given in HIGHPPM description.

Bit 4 – NEGPPM Negative PPM Correction
See CORRECTION and HIGHPPM field descriptions.
NEGPPM must be cleared to correct a crystal slower than 32.768 kHz.

Value	Name	Description
0	DISABLED	Positive correction (the divider will be slightly higher than 32768).
1	ENABLED	Negative correction (the divider will be slightly lower than 32768).

Bit 2 – UTC UTC Time Format
It is forbidden to write a one to the UTC and PERSIAN bits at the same time.

Value	Name	Description
0	DISABLED	Gregorian or Persian calendar.
1	ENABLED	UTC format.

Bit 1 – PERSIAN PERSIAN Calendar

Value	Name	Description
0	DISABLED	Gregorian calendar.
1	ENABLED	Persian calendar.

Bit 0 – HRMOD 12-/24-hour Mode

Value	Name	Description
0	24HOURS	24-hour mode is selected.
1	AMPM	12-hour mode is selected.

26.6.3 RTC Time Register

Name: RTC_TIMR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

In UTC mode, this register view is not relevant, see 26.6.7. RTC_TIMALR (UTC_MODE) .

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 22 – AMPM Ante Meridiem Post Meridiem Indicator
This bit is the AM/PM indicator in 12-hour mode.

Value	Description
0	AM.
1	PM.

Bits 21:16 – HOUR[5:0] Current Hour
The range that can be set is 1–12 (BCD) in 12-hour mode or 0–23 (BCD) in 24-hour mode.

Bits 14:8 – MIN[6:0] Current Minute
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – SEC[6:0] Current Second
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

26.6.4 RTC Time Register (UTC_MODE)

Name: RTC_TIMR (UTC_MODE)
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This configuration is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Current UTC Time
Any value can be set.

26.6.5 RTC Calendar Register

Name: RTC_CALR
Offset: 0x0C
Reset: 0x01E11220
Property: Read/Write

In UTC mode, values read in this register are not relevant.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	YEAR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 29:24 – DATE[5:0] Current Day in Current Month
The range that can be set is 01–31 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 23:21 – DAY[2:0] Current Day in Current Week
The range that can be set is 1–7 (BCD).
The coding of the number (which number represents which day) is user-defined as it has no effect on the date counter.

Bits 20:16 – MONTH[4:0] Current Month
The range that can be set is 01–12 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 15:8 – YEAR[7:0] Current Year
The range that can be set is 00–99 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – CENT[6:0] Current Century
The range that can be set is 20 (Gregorian) or 13–14 (Persian) (BCD).
The lowest four bits encode the units. The higher bits encode the tens.
Note: Value 20 (BCD) is always written in CENT whatever the value entered, thus there is no trigger event on RTC_VER.NVCAL regarding CENT.

26.6.6 RTC Time Alarm Register

Name: RTC_TIMALR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

To change one of the SEC, MIN, HOUR fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to RTC_TIMALR. The first access clears the enable corresponding to the field to change (SECEN, MINEN, HOUREN). If the field is already cleared, this access is not required. The second access performs the change of value (SEC, MIN, HOUR). The third access is required to re-enable the field by writing 1 in the SECEN, MINEN, HOUREN fields.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HOUREN	AMPM	HOUR[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MINEN	MIN[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SECEN	SEC[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 23 – HOUREN Hour Alarm Enable

Value	Name	Description
0	DISABLED	The hour-matching alarm is disabled.
1	ENABLED	The hour-matching alarm is enabled.

Bit 22 – AMPM AM/PM Indicator
This field is the alarm field corresponding to the BCD-coded hour counter.

Bits 21:16 – HOUR[5:0] Hour Alarm
This field is the alarm field corresponding to the BCD-coded hour counter.

Bit 15 – MINEN Minute Alarm Enable

Value	Name	Description
0	DISABLED	The minute-matching alarm is disabled.
1	ENABLED	The minute-matching alarm is enabled.

Bits 14:8 – MIN[6:0] Minute Alarm
This field is the alarm field corresponding to the BCD-coded minute counter.

Bit 7 – SECEN Second Alarm Enable

Value	Name	Description
0	DISABLED	The second-matching alarm is disabled.
1	ENABLED	The second-matching alarm is enabled.

Bits 6:0 – SEC[6:0] Second Alarm
This field is the alarm field corresponding to the BCD-coded second counter.

26.6.7 RTC Time Alarm Register (UTC_MODE)

Name: RTC_TIMALR (UTC_MODE)
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This configuration is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] UTC_TIME Alarm
This field is the alarm field corresponding to the UTC time counter. To change it, proceed as follows:
1. Disable the UTC alarm by clearing RTC_CALALR.UTCEN if it is not already cleared.
2. Change the UTC_TIME alarm value.
3. Enable the UTC alarm by setting RTC_CALALR.UTCEN.

26.6.8 RTC Calendar Alarm Register

Name: RTC_CALALR
Offset: 0x14
Reset: 0x01010000
Property: Read/Write

In UTC mode, this register view is not relevant, see 26.6.9. RTC_CALALR (UTC_MODE).

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

To change one of the DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to RTC_CALALR. The first access clears the enable corresponding to the field to change (DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (DATE, MONTH). The third access is required to re-enable the field by writing 1 in DATEEN, MTHEN fields.

Bit	31	30	29	28	27	26	25	24
	DATEEN		DATE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	MTHEN			MONTH[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – DATEEN Date Alarm Enable

Value	Name	Description
0	DISABLED	The date-matching alarm is disabled.
1	ENABLED	The date-matching alarm is enabled.

Bits 29:24 – DATE[5:0] Date Alarm
This field is the alarm field corresponding to the BCD-coded date counter.

Bit 23 – MTHEN Month Alarm Enable

Value	Name	Description
0	DISABLED	The month-matching alarm is disabled.
1	ENABLED	The month-matching alarm is enabled.

Bits 20:16 – MONTH[4:0] Month Alarm
This field is the alarm field corresponding to the BCD-coded month counter.

26.6.9 RTC Calendar Alarm Register (UTC_MODE)

Name: RTC_CALALR (UTC_MODE)
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								UTCEN
Access								R/W
Reset								0

Bit 0 – UTCEN UTC Alarm Enable

Value	Name	Description
0	DISABLED	The UTC-matching alarm is disabled.
1	ENABLED	The UTC-matching alarm is enabled.

26.6.10 RTC Status Register

Name: RTC_SR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Free Running Error
If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	CORRECT	The internal free running counters are carrying valid values since the last read of the Status register (RTC_SR).
1	ERR_TIMEDATE	The internal free running counters have been corrupted (invalid date or time, non-BCD values) since the last read and/or they are still invalid.

Bit 4 – CALEV Calendar Event
The calendar event is selected in RTC_CR.TIMEVSEL and can be any one of the following events: week change, month change and year change. If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_CALEVENT	No calendar event has occurred since the last clear.
1	CALEVENT	At least one calendar event has occurred since the last clear.

Bit 3 – TIMEV Time Event
The time event is selected in RTC_CR.TIMEVSEL and can be any one of the following events: minute change, hour change, noon, midnight (day change). If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_TIMEVENT	No time event has occurred since the last clear.
1	TIMEVENT	At least one time event has occurred since the last clear.

Bit 2 – SEC Second Event

Value	Name	Description
0	NO_SECEVENT	No second event has occurred since the last clear.
1	SECEVENT	At least one second event has occurred since the last clear.

Bit 1 – ALARM Alarm Flag

Value	Name	Description
0	NO_ALARMEVENT	No alarm matching condition occurred.
1	ALARMEVENT	An alarm matching condition has occurred.

Bit 0 – ACKUPD Acknowledge for Update

Value	Name	Description
0	FREERUN	Time and calendar registers cannot be updated.
1	UPDATE	Time and calendar registers can be updated.

26.6.11 RTC Status Clear Command Register

Name: RTC_SCCR
Offset: 0x1C
Reset: –
Property: Write-only

To avoid missing clearing commands, wait for three slow clock cycles between two accesses to this register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Clears the corresponding status flag in the Status register (RTC_SR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRCLR Time and/or Date Free Running Error Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALCLR Calendar Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMCLR Time Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECCLR Second Clear

Bit 1 – ALRCLR Alarm Clear

Bit 0 – ACKCLR Acknowledge Clear

26.6.12 RTC Interrupt Enable Register

Name: RTC_IER
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERREN Time and/or Date Error Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALEN Calendar Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMEN Time Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECEN Second Event Interrupt Enable

Bit 1 – ALREN Alarm Interrupt Enable

Bit 0 – ACKEN Acknowledge Update Interrupt Enable

26.6.13 RTC Interrupt Disable Register

Name: RTC_IDR
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRDIS Time and/or Date Error Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALDIS Calendar Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMDIS Time Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECDIS Second Event Interrupt Disable

Bit 1 – ALRDIS Alarm Interrupt Disable

Bit 0 – ACKDIS Acknowledge Update Interrupt Disable

26.6.14 RTC Interrupt Mask Register

Name: RTC_IMR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CAL	TIM	SEC	ALR	ACK
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Error Mask
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CAL Calendar Event Interrupt Mask
If the RTC is configured in UTC mode, this bit is not relevant.

Bit 3 – TIM Time Event Interrupt Mask
If the RTC is configured in UTC mode, this bit is not relevant.

Bit 2 – SEC Second Event Interrupt Mask

Bit 1 – ALR Alarm Interrupt Mask

Bit 0 – ACK Acknowledge Update Interrupt Mask

26.6.15 RTC Valid Entry Register

Name: RTC_VER
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

If the RTC is configured in UTC mode, the values returned by this register are not relevant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					NVCALALR	NVTIMALR	NVCAL	NVTIM
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – NVCALALR Non-valid Calendar Alarm

Value	Description
0	No invalid data has been detected in RTC_CALALR (Calendar Alarm register).
1	RTC_CALALR has contained invalid data since it was last programmed.

Bit 2 – NVTIMALR Non-valid Time Alarm

Value	Description
0	No invalid data has been detected in RTC_TIMALR (Time Alarm register).
1	RTC_TIMALR has contained invalid data since it was last programmed.

Bit 1 – NVCAL Non-valid Calendar

Value	Description
0	No invalid data has been detected in RTC_CALR (Calendar register).
1	RTC_CALR has contained invalid data since it was last programmed.

Bit 0 – NVTIM Non-valid Time

Value	Description
0	No invalid data has been detected in RTC_TIMR (Time register).
1	RTC_TIMR has contained invalid data since it was last programmed.

26.6.16 RTC TimeStamp Time Register 0

Name: RTC_TSTR0
Offset: 0xB0
Reset: 0x00000000
Property: Read-only

These fields are valid for non-UTC mode only.

RTC_TSTR0 reports the timestamp of the first tamper event after reading RTC_TSSR0.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR0)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_TSSR0)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no more possible to know the exact number of tamper events.
If this field is not null, this implies that at least one tamper event occurs since last register reset and that the values stored in timestamping registers are valid.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_TSSR0)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_TSSR0)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_TSSR0)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_TSSR0)

26.6.17 RTC TimeStamp Time Register 0 (UTC_MODE)

Name: RTC_TSTR0 (UTC_MODE)
Offset: 0xB0
Reset: 0x00000000
Property: Read-only

RTC_TSTR0 reports the timestamp of the first tamper event after reading RTC_TSSR0.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR0)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_TSSR0)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no more possible to know the exact number of tamper events.
If this field is not null, this implies that at least one tamper event occurs since last register reset and that the values stored in timestamping registers are valid.

26.6.18 RTC TimeStamp Time Register 1

Name: RTC_TSTR1
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

These fields are valid for non-UTC mode only.

RTC_TSTR1 reports the timestamp of the last tamper event after reading RTC_TSSR1.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR1)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_TSSR1)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_TSSR1)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_TSSR1)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_TSSR1)

26.6.19 RTC TimeStamp Time Register 1 (UTC_MODE)

Name: RTC_TSTR1 (UTC_MODE)
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

RTC_TSTR1 reports the timestamp of the last tamper event after reading RTC_TSSR1.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR1)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

26.6.20 RTC TimeStamp Date Register

Name: RTC_TSDRx
Offset: 0xB4 + x*0x0C [x=0..1]
Reset: 0x00000000
Property: Read-only

These fields contain the date and the source of a tamper occurrence if RTC_TSTR0.TEVCNT field is not null.

These fields are relevant for non-UTC mode only.

RTC_TSDR0 reports the timestamp of the first tamper event after reading RTC_TSSR0, and RTC_TSDR1 reports the timestamp of the last tamper event.

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YEAR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 29:24 – DATE[5:0] Date of the Tamper (cleared by reading RTC_TSSRx)

Bits 23:21 – DAY[2:0] Day of the Tamper (cleared by reading RTC_TSSRx)

Bits 20:16 – MONTH[4:0] Month of the Tamper (cleared by reading RTC_TSSRx)

Bits 15:8 – YEAR[7:0] Year of the Tamper (cleared by reading RTC_TSSRx)

Bits 6:0 – CENT[6:0] Century of the Tamper (cleared by reading RTC_TSSRx)

26.6.21 RTC TimeStamp Date Register (UTC_MODE)

Name: RTC_TSDRx (UTC_MODE)
Offset: 0xB4 + x*0x0C [x=0..1]
Reset: 0x00000000
Property: Read-only

RTC_TSDR0 reports the timestamp of the first tamper event after reading RTC_TSSR0, and RTC_TSDR1 reports the timestamp of the last tamper event.
This register is cleared by reading RTC_TSSRx.

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Time of the Tamper (UTC format)
This configuration is relevant only if UTC = 1 in RTC_MR.

26.6.22 RTC TimeStamp Source Register

Name: RTC_TSSRx
Offset: 0xB8 + x*0x0C [x=0..1]
Reset: 0x00000000
Property: Read-only

This register is cleared after read and the read access also performs a clear on RTC_TSTRx and RTC_TSDRx. The following configuration values are valid for all listed bit names of this register:

0: No alarm generated since the last clear.

1: An alarm has been generated by the corresponding monitor since the last clear.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			R	R	R	R		
Reset			0	0	0	0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					JTAG	TST		DWDTSW
Access					R	R		R
Reset					0	0		0

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector (cleared on read)

Bit 3 – JTAG JTAG Pins Monitor (cleared on read)

Bit 2 – TST Test Pin Monitor (cleared on read)

Bit 0 – DWDTSW Dual WatchDog Timer Event (cleared on read)

27. Shutdown Controller (SHDWC)

27.1 Description

The Shutdown Controller (SHDWC) controls the VDDIO and VDDCORE power supplies and the wake-up detection on debounced input lines.

27.2 Embedded Characteristics

- Shutdown Logic
  - Software assertion of the Shutdown Output Pin (SHDN)
  - Programmable de-assertion from wake-up events
- Wake-Up Logic
  - Programmable wake-up event detection through WKUP input pins and internal events (RTC, RTT and Security Module)
- Low-Power Logic
  - Automatic/manual control of the LPM output pin

27.3 Block Diagram

Figure 27-1. SHDWC Block Diagram

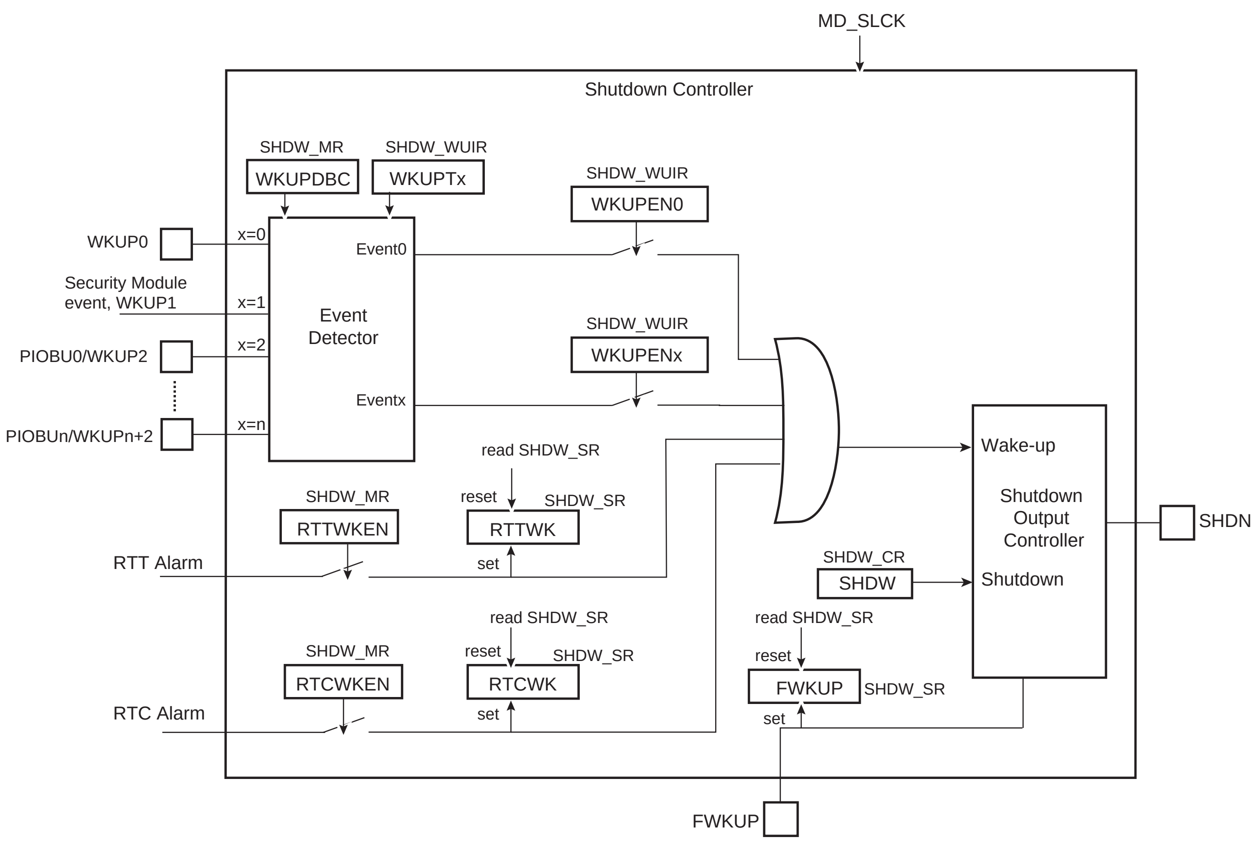
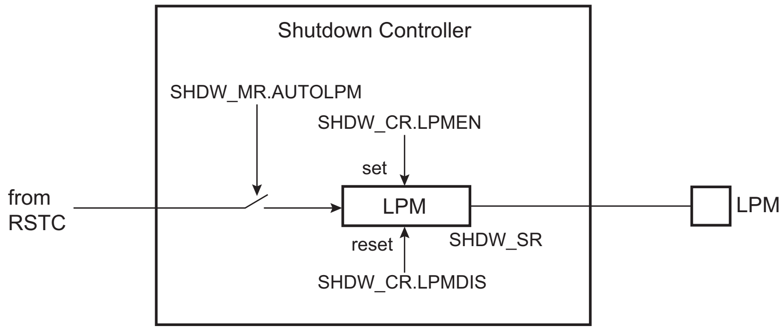


Figure 27-2. LPM Pad Management



27.4 I/O Lines Description

Table 27-1. I/O Lines Description

Name	Description	Type
WKUP0	Wake-up inputs	Input
PIOBU0-3	Wake-up inputs, WKUP(2-5)	Input
SHDN	Shutdown output	Output
LPM	Low-Power mode	Output

27.5 Product Dependencies

27.5.1 Power Management

The SHDWC is continuously clocked by the Monitoring Domain Slow Clock (MD_SLCK). The Power Management Controller has no effect on the behavior of the SHDWC.

27.6 Functional Description

The SHDWC manages the main power supply. To do so, it is supplied with VDDBU and manages wake-up input pins and one output pin, SHDN.

A typical application connects the pin SHDN to the enable input of the device's power supply circuit. The wake-up inputs (WKUPn) connect to any push-buttons or signal that wake up the system.

The software is able to control the pin SHDN by writing the Shutdown Control Register (SHDW_CR) with the bit SHDW at 1. The shutdown is taken into account only two MD_SLCK cycles after the write of SHDW_CR. This register is password-protected and so the value written should contain the correct key for the command to be taken into account. As a result, the SHDN pin is driven low and the system should be powered down.

The SHDWC manages the LPM pin which is used to indicate to an external power supply that the drive of the system can be reduced. This pin can be configured to be automatically managed by software or by hardware.

27.6.1 Wake-up Inputs

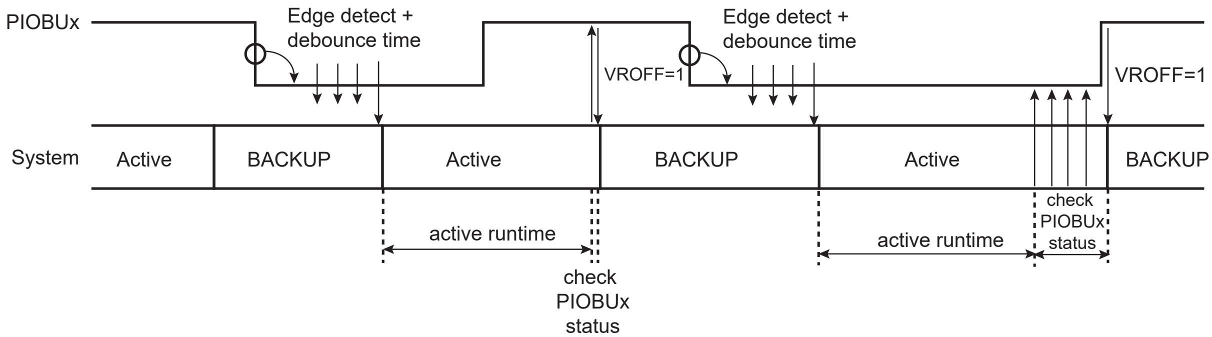
Any level change on a PIOBUx, WKUP pin, or Security Module event, can trigger a wake-up. Wake-up is configured in the Mode register (SHDW_MR) and Wakeup Inputs register (SHDW_WUIR). The transition detector can be programmed to detect either a positive or negative transition on any PIOBUx, WKUP pin. The detection can also be disabled. Programming is performed by enabling the Wake-up Input (WKUPENx bit) and defining the Wake-up Input Type (WKUPTx bit) in the SHDW_WUIR.

Moreover, a debouncing circuit can be programmed for PIOBUx, WKUP. The debouncing circuit filters pulses on PIOBUx, WKUP shorter than the programmed value in SHDW_MR.WKUPDBC. If the programmed level change is detected on a pin, a counter starts. When the counter reaches the value programmed in WKUPDBC, the SHDN pin is released. If a new input change is detected before the counter reaches the corresponding value, the counter is stopped and cleared. One counter is shared among all PIOBUx, WKUP inputs and all programmed level detection is merged into this counter. The WKUPISx bit of the Status register (SHDW_SR) reports the detection of the programmed events on PIOBUx, WKUP with a reset after the read of SHDW_SR.

Figure 27-3. Entering and Exiting Backup Mode with a PIOBUx, WKUP Pin

WKUPDBC > 0

WKUPTx=0



The SHDWC can be programmed so as to activate the wake-up using the RTC and RTT alarms or the Security Module event (detection of the rising edge event is synchronized with MD_SLCK). This is done by writing SHDW_MR.RTCWKEN or RTTWKEN, or by writing the SHDW_WUIR.WKUPEN1 (Security Module event is connected on the WKUP1 wake-up input). When enabled, the detection of the RTC or RTT alarms or the Security Module event is reported in SHDW_SR.RTCWK, RTTWK or WKUPIS1. These bits are cleared after reading SHDW_SR. When using the RTC or RTT alarms or the Security Module event to wake up the system, the user must ensure that the RTC and RTT alarm status flags and the Security Module event flag are cleared before shutting down the system. Otherwise, no rising edge of the status flags may be detected and the wake-up fails

27.6.2 Low-Power Mode Pin Control

The SHDWC manages the LPM pin which is used to indicate to an external power supply to enter or exit a special powering state. The LPM pin can be configured by software or by hardware.

To configure LPM with software:

- Set SHDWC_CR.LPMEN to set LPM 'high' (VBAT level)
- Set SHDWC_CR.LPMDIS to set LPM 'low' (VBAT level)

Writing both SHDW_CR.LPMEN and SHDW_CR.LPMDIS has no effect.

For LPM configuration via hardware, SHDW_MR.AUTOLPM must be set to '1'. The configuration is done automatically by the SHDWC and the RSTC when entering and exiting ULP mode 2:

- The LPM pin is set 'high' when the system enters ULP mode 2
- The LPM pin is set 'low' when the system exits ULP mode 2

By default, at the first power-up, LPM starts at low-level.

27.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SHDW_CR	31:24	KEY[7:0]							
		23:16		LPMDIS	LPMEN					
		15:8								
		7:0								SHDW
0x04	SHDW_MR	31:24						WKUPDBC[2:0]		
		23:16				AUTOLPM			RTCWKEN	RTTWKEN
		15:8								
		7:0								
0x08	SHDW_SR	31:24								
		23:16			WKUPIS5	WKUPIS4	WKUPIS3	WKUPIS2	WKUPIS1	WKUPIS0
		15:8								LPM
		7:0			RTCWK	RTTWK				WKUPS
0x0C	SHDW_WUIR	31:24								
		23:16			WKUPT5	WKUPT4	WKUPT3	WKUPT2	WKUPT1	WKUPT0
		15:8								
		7:0			WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1	WKUPEN0

27.7.1 SHDWC Control Register

Name: SHDW_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
		LPMDIS	LPMEN					
Access		W	W					
Reset		–	–					

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SHDW
Access								W
Reset								–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 22 – LPMDIS LPM Pad Disable

Value	Description
0	No effect.
1	The LPM pad is set low (external regulator is set in standard powering state).

Bit 21 – LPMEN LPM Pad Enable

Value	Description
0	No effect.
1	The LPM pad is set high (external regulator is set in special powering state).

Bit 0 – SHDW Shutdown Command

Value	Description
0	No effect.
1	If KEY value is correct, asserts the SHDN pin.

27.7.2 SHDWC Mode Register

Name: SHDW_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
						WKUPDBC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
				AUTOLPM			RTCWKEN	RTTWKEN
Access				W			R/W	R/W
Reset				0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 26:24 – WKUPDBC[2:0] Wake-up Inputs Debouncer Period

Value	Name	Description
0	IMMEDIATE	Immediate, no debouncing, detected active at least on one MD_SLCK edge
1	3_SLCK	PIOBUx shall be in its active state for at least 3 MD_SLCK periods
2	32_SLCK	PIOBUx shall be in its active state for at least 32 MD_SLCK periods
3	512_SLCK	PIOBUx shall be in its active state for at least 512 MD_SLCK periods
4	4096_SLCK	PIOBUx shall be in its active state for at least 4,096 SLCK periods
5	32768_SLCK	PIOBUx shall be in its active state for at least 32,768 MD_SLCK periods

Bit 20 – AUTOLPM Automatic LPM Pad Management

Value	Description
0	The LPM pad is never automatically modified by the RSTC.
1	The LPM pad is automatically modified by the RSTC when the system goes into or out of the ULP mode 2.

Bit 17 – RTCWKEN Real-time Clock Wake-up Enable

Value	Description
0	The RTC Alarm signal has no effect on the SHDWC.
1	The RTC Alarm signal forces the de-assertion of the SHDN pin.

Bit 16 – RTTWKEN Real-time Timer Wake-up Enable

Value	Description
0	The RTT Alarm signal has no effect on the SHDWC.
1	The RTT Alarm signal forces the de-assertion of the SHDN pin.

27.7.3 SHDWC Status Register

Name: SHDW_SR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Note: The events are detected only when the system is in Backup mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			WKUPIS5	WKUPIS4	WKUPIS3	WKUPIS2	WKUPIS1	WKUPIS0
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
								LPM
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
			RTCWK	RTTWK				WKUPS
Access			R	R				R
Reset			0	0				0

Bits 16, 17, 18, 19, 20, 21 – WKUPISx Wake-up x Input Status

Value	Name	Description
0	DISABLE	The corresponding wake-up input is disabled, or was inactive at the time the debouncer triggered a wake-up event.
1	ENABLE	The corresponding wake-up input was active at the time the debouncer triggered a wake-up event.

Bit 8 – LPM Low-Power Mode Pad Status

Value	Description
0	The LPM pad is currently set to 0.
1	The LPM pad is currently set to 1.

Bit 5 – RTCWK Real-time Clock Wake-up

Value	Description
0	No wake-up alarm from the RTC occurred since the last read of SHDW_SR.
1	At least one wake-up alarm from the RTC occurred since the last read of SHDW_SR.

Bit 4 – RTTWK Real-time Timer Wake-up

Value	Description
0	No wake-up alarm from the RTT occurred since the last read of SHDW_SR.
1	At least one wake-up alarm from the RTT occurred since the last read of SHDW_SR.

Bit 0 – WKUPS PIOBU, WKUP Wake-up Status

Value	Name	Description
0	NO	No wake-up due to the assertion of the PIOBU, WKUP pins has occurred since the last read of SHDW_SR.
1	PRESENT	At least one wake-up due to the assertion of the PIOBU, WKUP pins has occurred since the last read of SHDW_SR.

27.7.4 SHDWC Wake-up Inputs Register

Name: SHDW_WUIR
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			WKUPT5	WKUPT4	WKUPT3	WKUPT2	WKUPT1	WKUPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1	WKUPEN0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21 – WKUPTx Wake-up Input x Type
As the Security Module event is connected to the WKUP1 wake-up input, WKUPT1 must be set to 1.

Value	Description
0	A falling edge followed by a low level on the corresponding wake-up input, for a period defined by WKUPDBC, forces wake-up of the core power supply.
1	A rising edge followed by a high level on the corresponding wake-up input, for a period defined by WKUPDBC, forces wake-up of the core power supply.

Bits 0, 1, 2, 3, 4, 5 – WKUPENx Wake-up Input x Enable

Value	Description
0	The corresponding wake-up input has no wake-up effect.
1	The corresponding wake-up input forces wake-up of the core power supply.

28. 64-bit Periodic Interval Timer (PIT64B)

28.1 Description

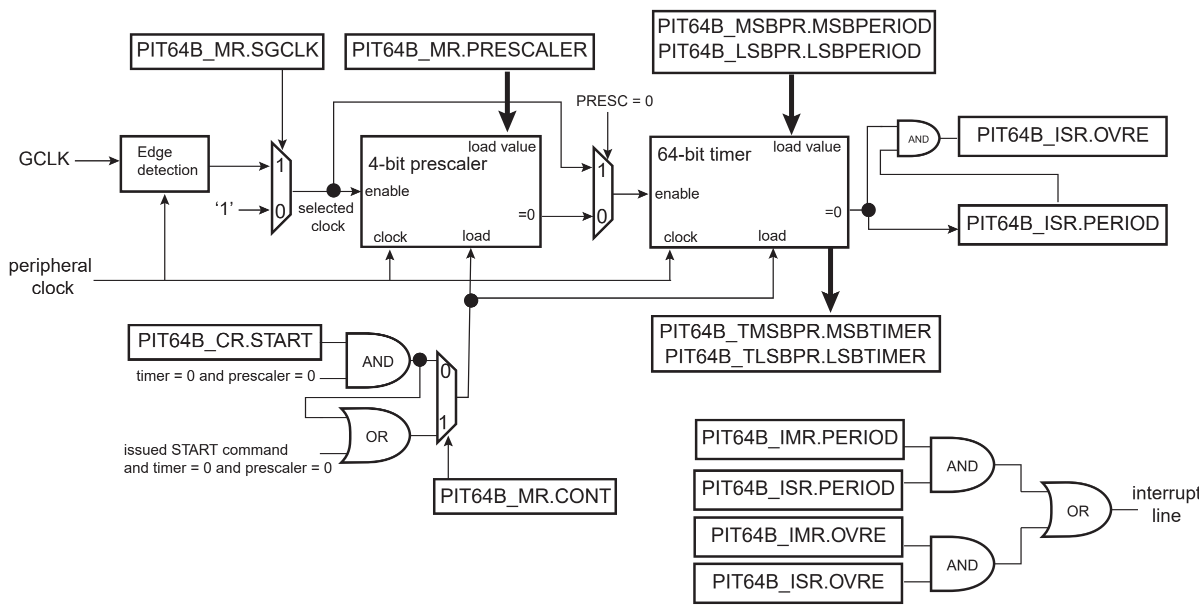
The 64-bit Periodic Interval Timer (PIT64B) provides the operating system scheduler interrupt, as well as any periodic source of interrupt to software. It is designed to offer maximum accuracy and efficient management, even for systems with long response times.

28.2 Embedded Characteristics

- 4-bit Prescaler
- 64-bit Timer
- Single Shot or Continuous Mode
- Safety/Security Access Reports
- Register Write protection

28.3 Block Diagram

Figure 28-1. Block Diagram



28.4 Product Dependencies

28.4.1 Power Management

The Power Management Controller (PMC) controls the PIT64B clock in order to save power. The programmer must first enable the PIT64B clock in the PMC before using the PIT64B.

After a hardware reset, the PIT64B clock is disabled by default.

28.4.2 Interrupt Generation

The PIT64B interface has an interrupt line connected to the Interrupt Controller.

Handling the PIT64B interrupt requires programming the Interrupt Controller before configuring the PIT64B.

28.5 Functional Description

28.5.1 Timer Clock Source

The two clock sources for the 64-bit timer are the peripheral clock and the generic clock (GCLK), which can be fully asynchronous to the peripheral clock. The selected clock can be prescaled before triggering the 64-bit timer.

The GCLK is selected as source clock for the prescaler when the SGCLK bit, in the Mode register (PIT64B_MR), is written to 1. The prescaler is active as soon as PIT64B_MR.PRESCALER>0.

If PIT64B_MR.PRESCALER=0, the timer is triggered either on each rising edge detection event of the GCLK if PIT64B_MR.SGCLK is written to 1, or on each rising edge of the peripheral clock.

If GCLK is selected, the frequency must be at least 3 times lower than the peripheral clock.

28.5.2 Single Period Mode

When the PIT64B_MR.CONT bit is written to 0, the PIT64B produces a single timer event. The timer period starts as soon as the START bit, in the Control register (PIT64B_CR), is written to 1. The period is defined by configuring the LSBPERIOD field in the LSB Period register (PIT64B_LSBPR) and the MSBPERIOD field in the MSB Period register (PIT64B_MSBPR). When the START command is issued, the 64-bit timer loads 0 and increments up to LSBPERIOD and MSBPERIOD field value minus 1, then automatically reloads 0 and stops.

When time reaches the maximum value, the PERIOD flag, in the Interrupt Status register (PIT64B_ISR), is set. No other period will be started until a new START command is issued.

After a period is started and while it is not elapsed, any new values written in PIT64B_MR, PIT64B_LSBPR or PIT64B_MSBPR have no effect on the current period if bit PIT64B_MR.SMOD=0 (see Figure Single Waveform in Single Period Mode if bit PIT64B_MR.SMOD=0

If PIT64B_MR.SMOD=1 a start is also performed as soon as PIT64B_LSBPR is written, thus a modification of the period can be performed on-the-fly with a single write operation if the period requires no more than 32 bits. When writing a 64-bit value, the 32-bit MSB part must be configured first, followed by the 32-bit LSB part (see Figure Waveform in Single Period Mode if bit PIT64B_MR.SMOD=1). When configuring a value lower or equal to 32 bits after processing a period defined on 64 bits, first PIT64B_MSBPR must be written to 0, and then the 32-bit LSB must be written into PIT64B_LSBPR.

If PIT64B_CR.SWRST is written to 1, the current period is immediately stopped.

Figure 28-2. Single Waveform in Single Period Mode if bit PIT64B_MR.SMOD=0

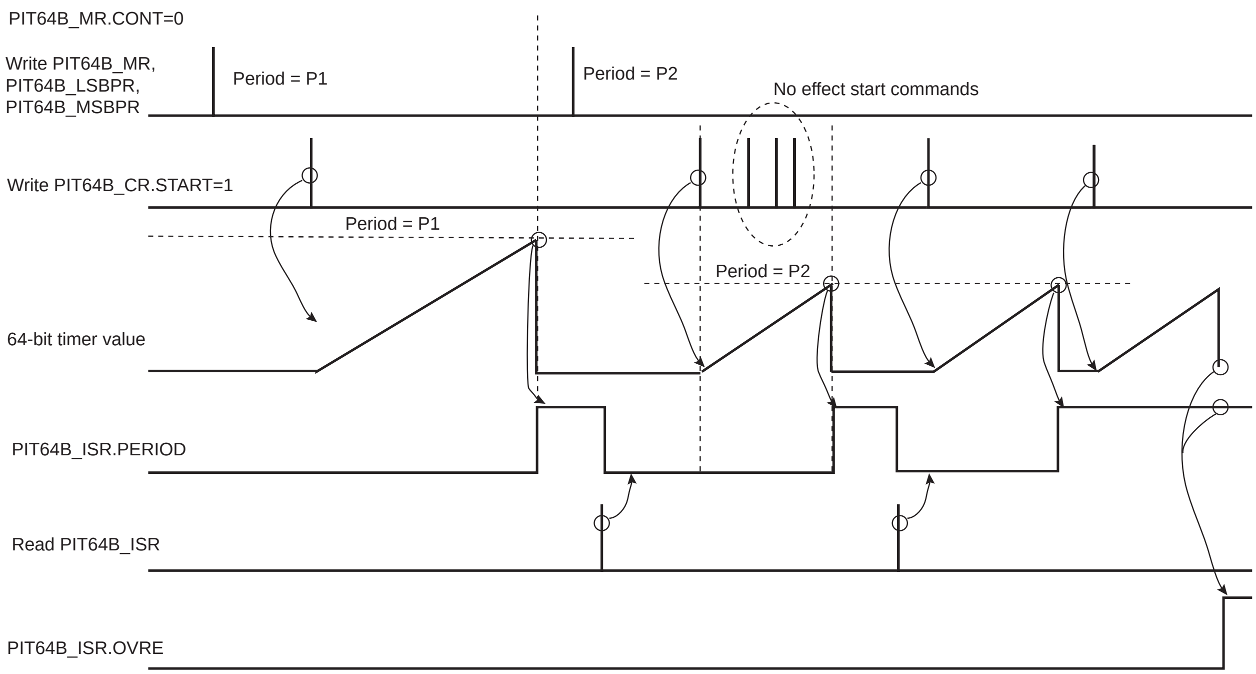
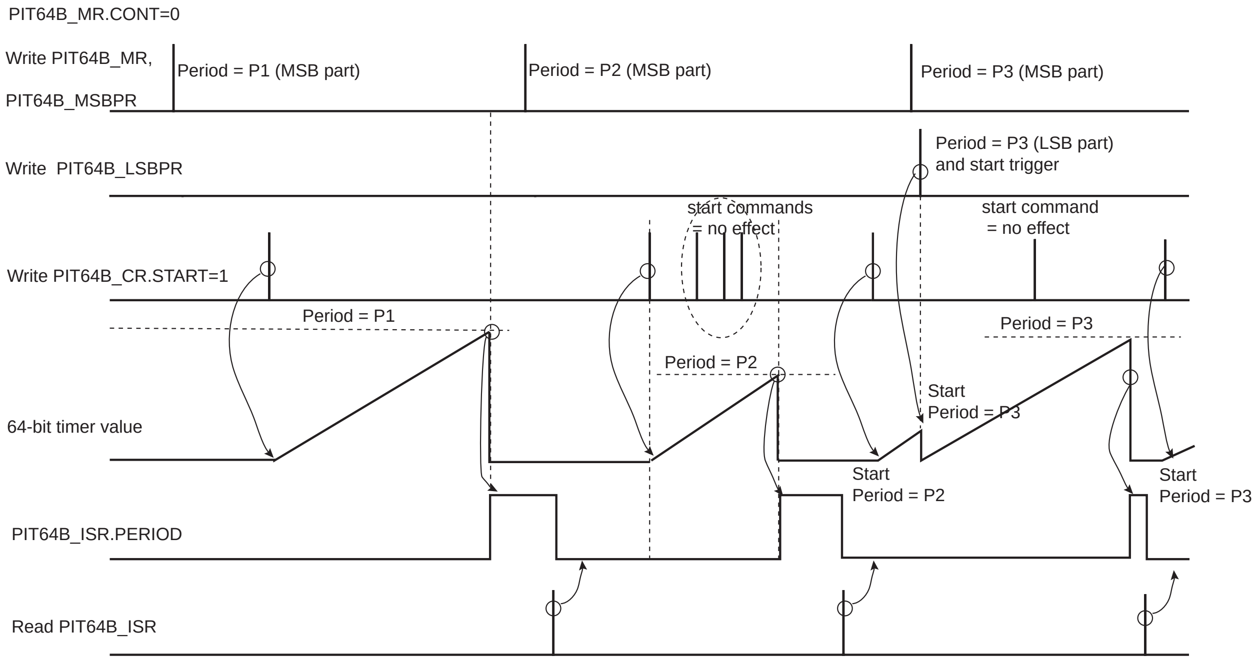


Figure 28-3. Waveform in Single Period Mode if bit PIT64B_MR.SMOD=1



28.5.3 Continuous Period Mode

When the PIT64B_MR.CONT bit is written to 1, the PIT64B continuously produces timer events. The timer is started as soon as the PIT64B_CR.START bit is written to 1. The period is defined by configuring the PIT64B_LSBPR.LSBPERIOD field and PIT64B_MSBPR.MSBPERIOD field. When the START command is issued, the 64-bit timer loads 0 and increments up to LSBPERIOD and MSBPERIOD field value minus 1, then automatically reloads 0 anc restarts a new counting period until bit PIT64B_CR.SWRST is written to 1.

When the timer reaches its maximum value, the flag PIT64B_ISR.PERIOD is set. PIT64B_ISR.PERIOD is cleared when reading PIT64B_ISR. If a new period elapses and the PIT64B_ISR.PERIOD is 1, the PIT64B_ISR.OVRE flag is set to indicate a potential latency at system level

After the START command has been issued, any new values written in PIT64B_MR, PIT64B_LSBPR or PIT64B_MSBPR have no effect on the current period if bit PIT64B_MR.SMOD=0 (see Figure Waveform in Continuous Period Mode if the bit PIT64B_MR.SMOD=0). A software reset must be issued before configuring new values in PIT64B_LSBPR and PIT64B_MSBPR if bit PIT64B_MR.SMOD=0.

If PIT64B_MR.SMOD=1 a start can be also performed as soon as PIT64B_LSBPR is written, thus a modification of the period can be performed on-the-fly with a single write operation if the period requires no more than 32 bits. When writing a 64-bit value, the 32-bit MSB part must be configured first followed by a 32-bit LSB part (see Figure Waveform in Continuous Period Mode if the bit PIT64B_MR.SMOD=1). When configuring a value lower or equal to 32 bits after processing a period defined on 64 bits, first PIT64B_MSBPR must be written to 0, and then the 32-bit LSB must be written into PIT64B_LSBPR.

If PIT64B_CR.SWRST is written to 1, the current period is immediately stopped.

Figure 28-4. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=0

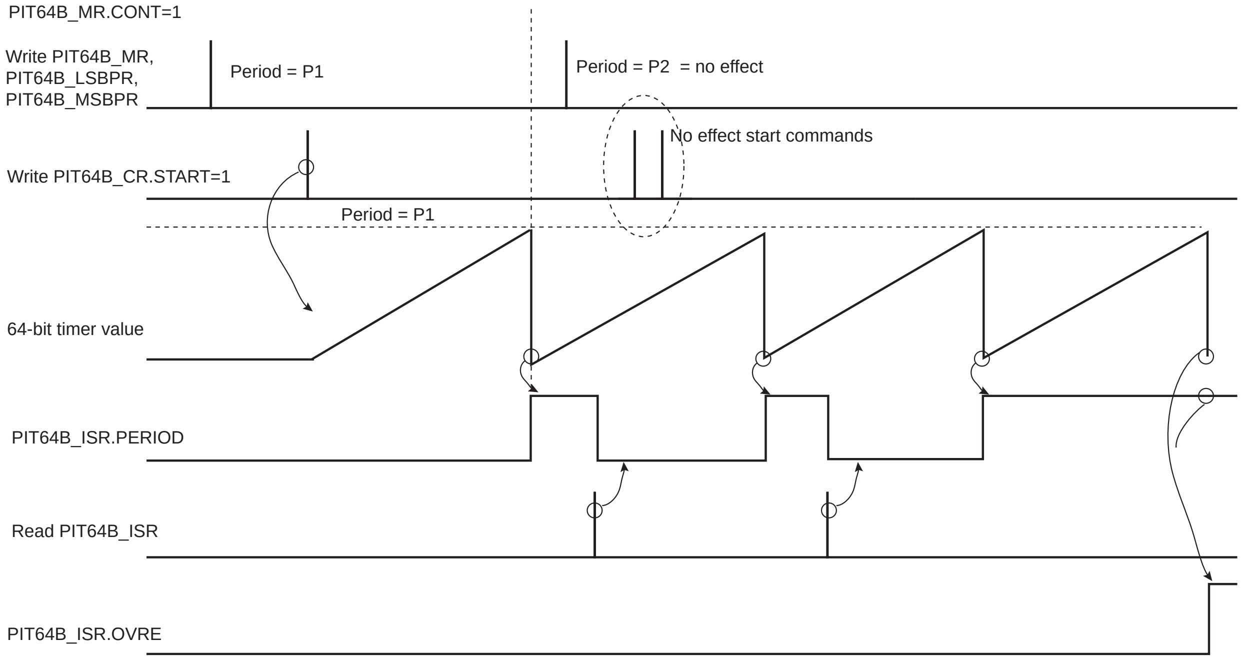
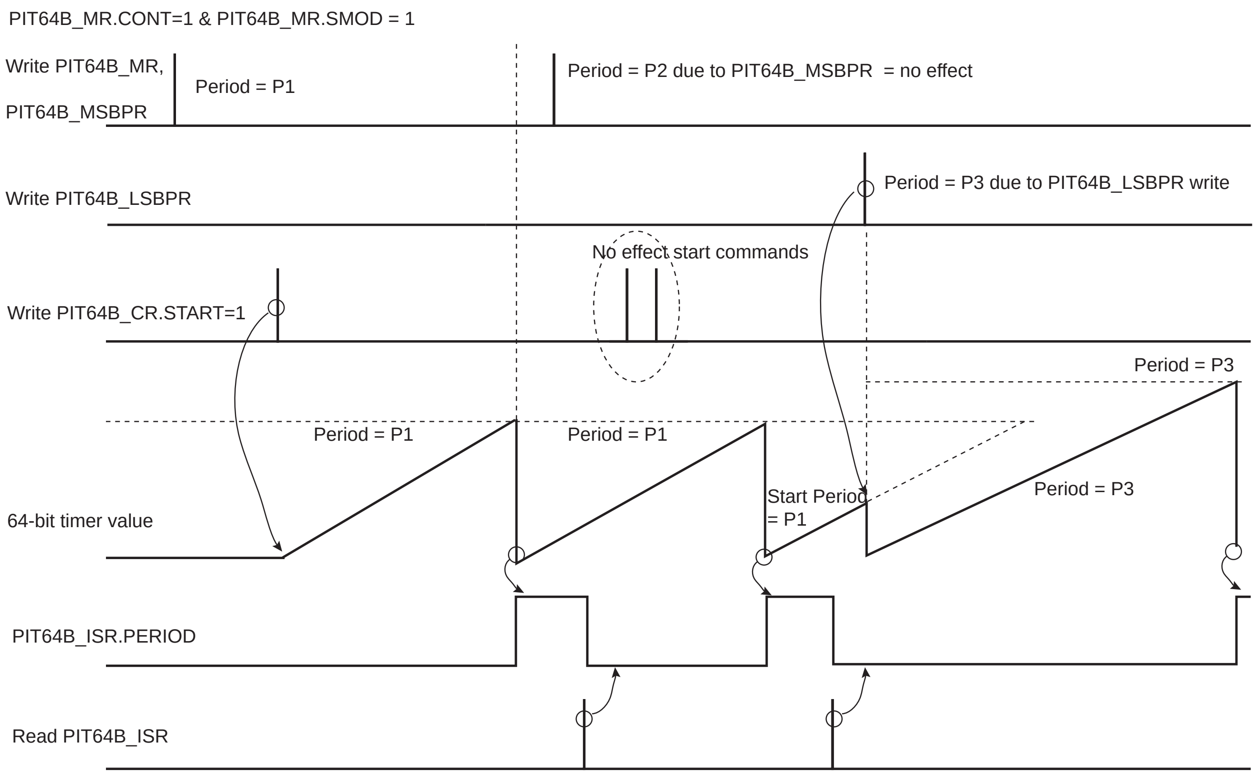


Figure 28-5. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=1



28.5.4 Security and Safety Analysis and Reports

Several types of checks are performed when the PIT64B is running.

The peripheral clock of the PIT64B is monitored by a specific circuitry to detect abnormal waveforms on the internal clock that may affect the behavior of the PIT64B. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the CGD flag is set in the Write Protection Status register (PIT64B_WPSR), an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal timer sequence of the PIT64B is also monitored and if an abnormal state is detected, the flag PIT64B_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the PIT64B are monitored and if an incorrect access is performed, the flag PIT64B_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the PIT64B_WPSR.SWETYP field (see PIT64B Write Protection Status Register for details), e.g., writing PIT64B_MR, PIT64B_LSBPR (if PIT64B_MR.SMOD=0), PIT64B_MSBPR (if PIT64B_MR.SMOD=0) when the timer is running (after a START command has been issued) is an error. PIT64B_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when PIT64B_WPSR is read.

If one of these flags is set, the PIT64B_ISR.SECE flag is set and can trigger an interrupt if the SECE bit, in the Interrupt Mask register (PIT64B_IMR), is '1'. SECE is cleared by reading PIT64B_ISR.

28.5.5 Register Write Protection

To prevent any single software error from corrupting PIT64B behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the PIT64B Write Protection Mode Register (PIT64B_WPMR).

If a write access to a write-protected register is detected, the WPVS (Write Protection Violation Status) flag in the PIT64B Write Protection Status Register (PIT64B_WPSR) is set and the WPVSRC (Write Protection Violation Source) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading PIT64B_WPSR.

The following registers can be write-protected when WPEN is set in PIT64B_WPMR:

- PIT64B Mode Register
- PIT64B LSB Period Register
- PIT64B MSB Period Register

Note: PIT64B LSB Period Register and PIT64B MSB Period Register are not write-protected if PIT64B_MR.SMOD=1.

The following registers can be write-protected when WPITEN is set in PIT64B_WPMR:

- PIT64B Interrupt Enable Register
- PIT64B Interrupt Disable Register

The following register can be write-protected when WPCREN is set in PIT64B_WPMR:

- PIT64B Control Register

28.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PIT64B_CR	31:24								
		23:16								
		15:8								SWRST
		7:0								START
0x04	PIT64B_MR	31:24								
		23:16								
		15:8					PRESCALER[3:0]			
		7:0				SMOD	SGCLK			CONT
0x08	PIT64B_LSBPR	31:24	LSBPERIOD[31:24]							
		23:16	LSBPERIOD[23:16]							
		15:8	LSBPERIOD[15:8]							
		7:0	LSBPERIOD[7:0]							
0x0C	PIT64B_MSBPR	31:24	MSBPERIOD[31:24]							
		23:16	MSBPERIOD[23:16]							
		15:8	MSBPERIOD[15:8]							
		7:0	MSBPERIOD[7:0]							
0x10	PIT64B_IER	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x14	PIT64B_IDR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x18	PIT64B_IMR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x1C	PIT64B_ISR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x20	PIT64B_TLSBR	31:24	LSBTIMER[31:24]							
		23:16	LSBTIMER[23:16]							
		15:8	LSBTIMER[15:8]							
		7:0	LSBTIMER[7:0]							
0x24	PIT64B_TMSBR	31:24	MSBTIMER[31:24]							
		23:16	MSBTIMER[23:16]							
		15:8	MSBTIMER[15:8]							
		7:0	MSBTIMER[7:0]							
0x28 ... 0xE3	Reserved									
0xE4	PIT64B_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	PIT64B_WPSR	31:24	ECLASS						SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

28.6.1 PIT64B Control Register

Name: PIT64B_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the PIT64B Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								–

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Performs a software reset, clears the configuration and stops any timer period in progress.

Bit 0 – START Start Timer

Value	Description
0	No effect.
1	The timer counter is started for 1 or more periods. If the START command is applied during a non-elapsed timer period, there is no effect. Thus, in Continuous mode, the SWRST command is the only command to stop the PIT64B. If PIT64B_MR.SMOD=1 a start is also performed as soon as PIT64B_LSBPR is written (see 28.5.2. Single Period Mode and 28.5.3. Continuous Period Mode).

28.6.2 PIT64B Mode Register

Name: PIT64B_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register.

When the timer is running, writing a value to this register has no effect. The value written is this register is loaded anytime before a START command is issued.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					PRESCALER[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				SMOD	SGCLK			CONT
Access				R/W	R/W			R/W
Reset				0	0			0

Bits 11:8 – PRESCALER[3:0] Prescaler Period

Value	Description
0	A prescaler divider of 1 is used.
1-15	The 64-bit timer is incremented at each (PRESCALER+1)x selected period (see SGCLK).

Bit 4 – SMOD Start Mode

Value	Description
0	Writing PIT64B_LSBPR does not start the timer period.
1	Writing PIT64B_LSBPR starts the timer period.

Bit 3 – SGCLK Generic Clock Selection Enable
If GCLK is asynchronous to the peripheral clock, a jitter of 1 peripheral clock period is created on the periodic interval event when Continuous mode is selected.

Value	Description
0	The prescaler is triggered at each rising edge of “Peripheral clock” and the timer is triggered.
1	GCLK clock is selected as clock source of the 8-bit prescaler.

Bit 0 – CONT Continuous Mode

Value	Description
0	A single period interrupt is generated from a START command.
1	Continuous periodic interrupts are generated after a single START command.

28.6.3 PIT64B LSB Period Register

Name: PIT64B_LSBPR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register or if PIT64B_MR.SMOD=1.

When the timer is running, if PIT64B_MR.SMOD=0, writing a value to this register has no effect. The value written is this register must be loaded anytime before a START command is issued if PIT64B_MR.SMOD=0. If PIT64B_MR.SMOD=1, a write access to this register restarts a timer period.

Bit	31	30	29	28	27	26	25	24
	LSBPERIOD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LSBPERIOD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LSBPERIOD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LSBPERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LSBPERIOD[31:0] 32 LSB of the Timer Period
This field defines the 32 LSB of the timer period. The timer period is defined by selected clock x {MSBPERIOD,LSBPERIOD}.

28.6.4 PIT64B MSB Period Register

Name: PIT64B_MSBPR
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register.

When the timer is running, if PIT64B_MR.SMOD=0, writing a value to this register has no effect. The value written is this register must be loaded anytime before a START command is issued if PIT64B_MR.SMOD=0.

Bit	31	30	29	28	27	26	25	24
	MSBPERIOD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MSBPERIOD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MSBPERIOD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MSBPERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSBPERIOD[31:0] 32 MSB of the Timer Period
This field defines the 32 MSB of the timer period. The timer period is defined by selected clock x {MSBPERIOD,LSBPERIOD}.

28.6.5 PIT64B Interrupt Enable Register

Name: PIT64B_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIT64B Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				W			W	W
Reset				–			–	–

Bit 4 – SECE Safety and/or Security Report Interrupt Enable

Bit 1 – OVRE Overrun Error Interrupt Enable

Bit 0 – PERIOD Elapsed Timer Period Interrupt Enable

28.6.6 PIT64B Interrupt Disable Register

Name: PIT64B_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIT64B Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				W			W	W
Reset				–			–	–

Bit 4 – SECE Safety and/or Security Report Interrupt Disable

Bit 1 – OVRE Overrun Error Interrupt Disable

Bit 0 – PERIOD Elapsed Timer Period Interrupt Disable

28.6.7 PIT64B Interrupt Mask Register

Name: PIT64B_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: Corresponding interrupt is not enabled.

1: Corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				R			R	R
Reset				0			0	0

Bit 4 – SECE Safety and/or Security Report Interrupt Mask

Bit 1 – OVRE Overrun Error Interrupt Mask

Bit 0 – PERIOD Elapsed Timer Period Interrupt Mask

28.6.8 PIT64B Interrupt Status Register

Name: PIT64B_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				R			R	R
Reset				0			0	0

Bit 4 – SECE Safety/Security Report (cleared on read)

Value	Description
0	There has been no security report in PIT64B_WPSR since the last read of PIT64B_ISR.
1	One security flag has been set in PIT64B_WPSR since the last read of PIT64B_ISR.

Bit 1 – OVRE Overrun Error (cleared on read)

Value	Description
0	No multiple rollovers occurred since the last read of PIT64B_ISR.
1	More than 1 rollover occurred since the last read of PIT64B_ISR.

Bit 0 – PERIOD Elapsed Timer Period Status Flag (cleared on read)

Value	Description
0	No timer rollover occurred since the last read of PIT64B_ISR.
1	A timer rollover occurred since the last read of PIT64B_ISR.

28.6.9 PIT64B Timer LSB Register

Name: PIT64B_TLSBR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	LSBTIMER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LSBTIMER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LSBTIMER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LSBTIMER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LSBTIMER[31:0] Current 32 LSB of the Timer
This field returns the 32 LSB of the current timer value.

28.6.10 PIT64B Timer MSB Register

Name: PIT64B_TMSBR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

When operating with a timer value greater than 32 bits (PIT64B_MSBPR.MSBPERIOD > 0), the PIT64B_TLSBR must be read first, followed by the read of PIT64B_TMSBR. This sequence generates an atomic read of the 64-bit timer value whatever the lapse of time between the accesses. When operating with a timer value up to 32 bits (PIT64B_MSBPR.MSBPERIOD=0), reading PIT64B_TMSBR is not required.

Bit	31	30	29	28	27	26	25	24
	MSBTIMER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSBTIMER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSBTIMER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBTIMER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSBTIMER[31:0] Current 32 MSB of the Timer
This field returns the 32 MSB of the current timer value.

28.6.11 PIT64B Write Protection Mode Register

Name: PIT64B_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504954	PASSWD	Writing any other value in this field aborts the write operation of the WPCREN, WPITEN and WPEN bits. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in PIT64B_WPSR.WPVSRC and the last software control error type is reported in PIT64B_WPSR.SWETYP. The PIT64B_ISR.SECE flag is set at the first error occurring within a series.
1	Only the first write protection violation source is reported in PIT64B_WPSR.WPVSRC and only the first software control error type is reported in PIT64B_WPSR.SWETYP. The PIT64B_ISR.SECE flag is set at the first error occurring within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Section 6.5 “Register Write Protection” for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

28.6.12 PIT64B Write Protection Status Register

Name: PIT64B_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS						SWETYP[1:0]	
Access	R						R	R
Reset	0						0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality.
1 (ERROR): A write access is performed into PIT64B_MR, PIT64B_LSBR, PIT64B_MSBR while the PIT64B is running.

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	UNDEF_RW	Access to an undefined address (warning).
3	WEIRD_ACTION	A write access is performed into PIT64B_MR, PIT64B_LSBR, PIT64B_MSBR while the PIT64B is running (abnormal).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of PIT64B_WPSR.
1	A software error has occurred since the last read of PIT64B_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of PIT64B_WPSR.

Value	Description
1	A peripheral internal sequencer error has occurred since the last read of PIT64B_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of PIT64B_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of PIT64B_WPSR. This flag can only be set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of the PIT64B_WPSR.
1	A write protection violation has occurred since the last read of the PIT64B_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

29. Chip Identifier (CHIPID)

29.1 Description

Chip Identifier (CHIPID) registers are used to recognize the device and its revision. These registers provide the sizes and types of the on-chip memories, as well as the set of embedded peripherals.

Two CHIPID registers are embedded: Chip ID register (CHIPID_CIDR) and Chip ID Extension register (CHIPID_EXID). Both registers contain a hard-wired value that is read-only.

The CHIPID_CIDR register contains the following fields:

- VERSION: Identifies the revision of the silicon
- ID: Product identifier
- EXT: Shows the use of the extension identifier register

The CHIPID_EXID register is device-dependent and reads ‘0’ if CHIPID_CIDR.EXT = 0.

29.2 Embedded Characteristics

- Chip ID Registers
  – Identification of the device revision, sizes of the embedded memories, set of peripherals, embedded processor

Table 29-1. CHIPID Registers

Device Name	CHIPID_CIDR	CHIPID_EXID
SAMA7G54	0x8016211x	0x00000000

29.3 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CHIPID_CIDR	31:24	EXT				ID[22:19]			
		23:16	ID[18:11]							
		15:8	ID[10:3]							
		7:0	ID[2:0]			1	VERSION[3:0]			
0x04	CHIPID_EXID	31:24	EXID[31:24]							
		23:16	EXID[23:16]							
		15:8	EXID[15:8]							
		7:0	EXID[7:0]							

29.3.1 Chip ID Register

Name: CHIPID_CIDR
Offset: 0x0
Reset: 0x8016211x
Property: Read-only

Values not listed for bitfields must be considered as “reserved”

Bit	31	30	29	28	27	26	25	24
	EXT				ID[22:19]			
Access	R				R	R	R	R
Reset	x				x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ID[18:11]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ID[10:3]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	ID[2:0]			1	VERSION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	x	1	1	2	6	1	0	8

Bit 31 – EXT Extension Flag

Value	Description
0	Chip ID has a single register definition without extension.
1	An extended Chip ID exists.

Bits 27:5 – ID[22:0] Product ID

Bit 4 – 1 Must be ‘1’.

Bits 3:0 – VERSION[3:0] Version of the Device
Current version of the device.

29.3.2 Chip ID Extension Register

Name: CHIPID_EXID
Offset: 0x4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EXID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EXID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – EXID[31:0] Chip ID Extension
This field is cleared if CHIPID_CIDR.EXT = 0.

30. OTP Memory Controller (OTPC)

30.1 Description

The OTP Memory Controller (OTPC) is the secure interface between the system and the OTP memory.

The default value of a memory bit is logic ‘0’ (not programmed). A programmed memory bit is logic ‘1’.

An OTP matrix is a type of non-volatile memory. Each bit in the matrix can be programmed only once. The bits are used to store data such as:

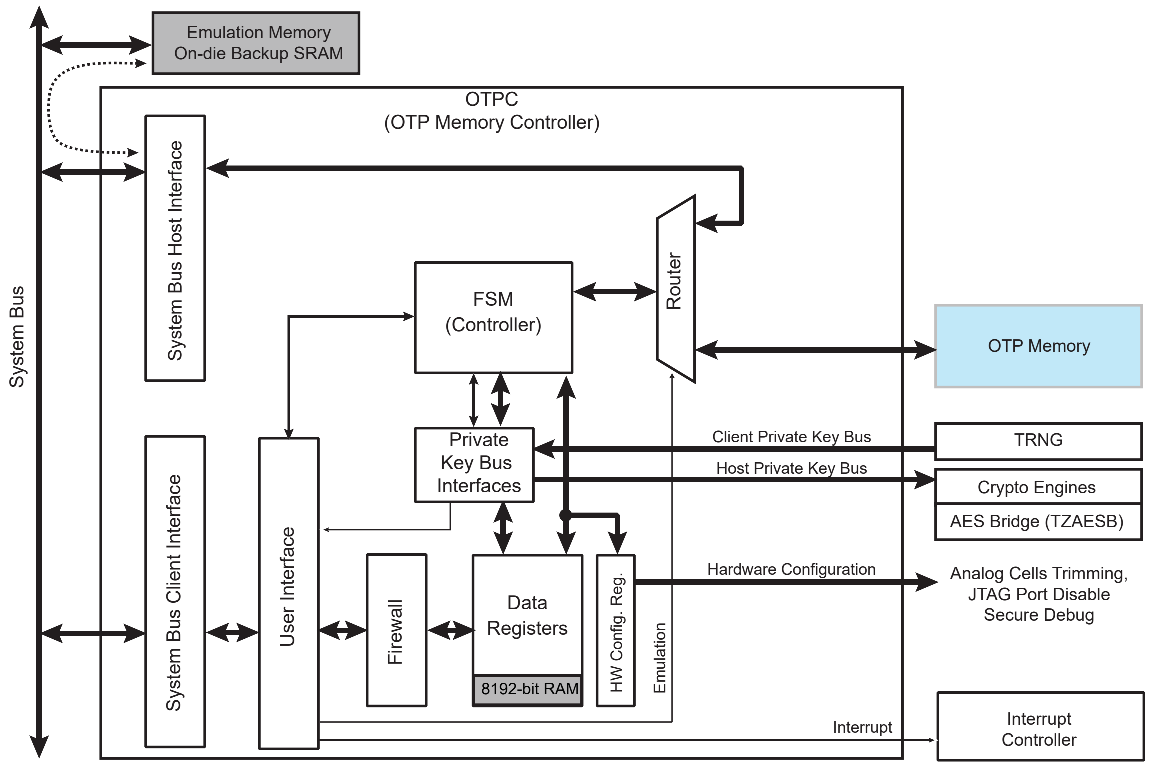
- calibration bits for analog cells (e.g. RC oscillators, etc.),
- hardware configuration settings (e.g. JTAG disable, etc.),
- chip identifiers,
- key data invisible by software,
- user data.

30.2 Embedded Characteristics

- Programs and Reads the Memory by Software
- Emulation Mode
- Automatic Check of Programmed Bits on Startup for Safe Operation
- User Area Organized by Packet for Flexibility on Size and Security:
  – Individual packet locking possibility (with checksum check)
  – Individual packet read access through only Private Key bus or System bus
  – Individual packet hiding (for packet with System bus access only)
  – Individual packet size of 32 bits to 8192 bits in 32-bit steps
  – Individual packet invalidation
- Firewall: Software/Hardware Protection Against Unexpected Read/Write

30.3 Block Diagram

Figure 30-1. OTPC Block Diagram



30.4 Product Dependencies

30.4.1 Power Management

The OTPC is clocked through the Power Management Controller (PMC), so the programmer must first to configure the PMC.

30.4.2 Interrupt Sources

The OTPC interface has an interrupt line connected to the Interrupt Controller.

Handling the OTPC interrupt requires programming the Interrupt Controller before configuring the OTPC.

30.5 Functional Description

30.5.1 Bus Interfaces

The OTPC features four bus interfaces to access the OTP memory:

- Host system bus
- Client system bus
- Host key bus (private key bus)
- Client key bus (private key bus)

The host system bus is used in Emulation mode to write and read data to/from the Emulation memory instead of the OTP memory. The host system bus can only access the emulation memory.

The client system bus is available to access to the user interface.

The host key bus is available to read keys stored in the User area of the OTP memory and transfer them to the client crypto-engines connected to this bus (e.g. AES, TZAESB). No data accessible to the host key bus are accessible to the System bus.

The client key bus is available to write some data to the User area of the OTP memory. No data coming from the client key bus are accessible to the system bus.

30.5.2 OTP Memory Partitioning

The OTP memory is partitioned into different areas:

- Reserved area
- 11-Kbyte User area

The initial value of the OTP memory is ‘0’ but the memory may contain some “defective” bits already set to the value ‘1’.

The memory is organized into 32-bit words.

30.5.3 User Area

30.5.3.1 Area Configuration and Control

The User area is controlled and configured through the OTPC Control (OTPC_CR), OTPC Mode (OTPC_MR) and OTPC Data (OTPC_DR) registers.

30.5.3.2 Area Mapping

The entire User area space is mapped into 32-bit words. Each 32-bit word is part of one packet.

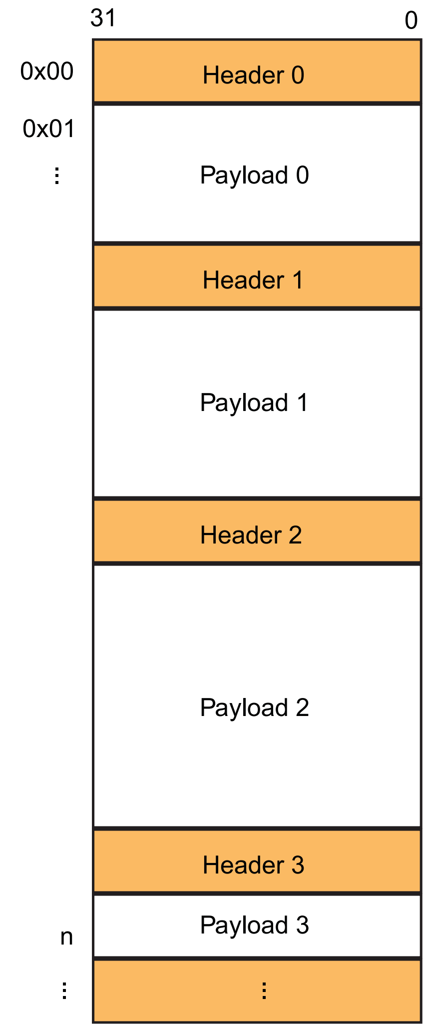
Each packet contains a 32-bit header and 32-bit words of payload (data). The number of payload words is defined in the header.

This mapping system allows three important functions for an OTP memory:

- Gives a flexible size for any type of data
- Improves security by identifying and masking key areas to the System bus,
- Provides an easy way to invalidate a packet and create a replacement packet (key, data) while free space is available in OTP.

The packet organization into the User area is shown in the figure below.

Figure 30-2. User Area Memory Mapping



The number of packets depends on the size of the User area and the size of each packet.

30.5.3.3 Packet Definition

Each packet contains:

- One 32-bit header field
- One payload field containing at least 32 bits of data and up to 256 x 32 bits of data (8192 bits)

The payload field has no effect on the hardware except for "special" packets.

30.5.3.3.1 Header Field

The following table provides the definition of the header content.

31	30	29	28	27	26	25	24
CHECKSUM							
23	22	21	20	19	18	17	16
CHECKSUM							
15	14	13	12	11	10	9	8
SIZE							
7	6	5	4	3	2	1	0

ONE	SECURE	INVLD	LOCK	PACKET

- PACKET: Indicates the packet type. Six types are available:
  - REGULAR packet (value 1) accessible through the User Interface
  - KEY packet (value 2) accessible only through the Private Key buses
  - BOOT_CONFIGURATION “special” packet (value 3)
  - SECURE_BOOT_CONFIGURATION “special” packet (value 4)
  - HARDWARE_CONFIGURATION “special” packet (value 5)
  - CUSTOM “special” packet (value 6)
- LOCK: Written by the controller when the checksum is generated.
- INVLD: Written when an invalidation process is requested.
- SECURE: Indicates if the packet must be available to the secure or non-secure domain. Also applies to the Key bus transfers.
- ONE: Must be written to ‘1’.
- SIZE: Indicates the size in 32-bit words of the payload field. SIZE = 0 means payload is 32-bit size, SIZE = 255 means payload is 8192-bit size.
  The entire packet size (in bits) in OTP memory is 32 (header) + (32 * (SIZE + 1)).
- CHECKSUM: Represents the checksum of the packet excluding the CHECKSUM field of the header. The real value of the CHECKSUM field is not readable; it is generated by the OTPC.
  - When CHECKSUM is read as 0, the checksum has not been generated. It is possible to modify the packet content.
  - When CHECKSUM is read as 0x33CC, the checksum has not been generated but some bits are already at 1. Locking the packet may fail.
  - When CHECKSUM is read as 0xA5A5, the checksum has been generated and the last check was successful. It is impossible to modify the packet content.
  - When CHECKSUM is read as 0xCC33, the header of the packet is corrupted.
  - When CHECKSUM is read as 0xFFFF, the entire packet is no longer valid. It is possible to modify the packet content and it is up to the software to program the payload to a different value if needed.
  - For all other values of CHECKSUM, the checksum has been generated and the last check failed to match the checksum written in the OTP memory. It is impossible to modify the packet content.

30.5.3.3.2 “Special” Packets

The payload of “special” packets is interpreted by the OTPC and some actions can be triggered while the “special” packets are read. The address of the “special” packets inside the area does not matter.

If the checksum has been generated and does not match during the last read, the payload is not interpreted by the OTPC.

The table below provides the list of “special” packets.

Table 30-1. “Special” Packets

Name	SIZE	PACKET	Description
Boot Configuration	(see Note 1)	3	Boot configuration
Secure Boot Configuration	(see Note 1)	4	Secure Boot configuration
User Hardware Configuration	1	5	Hardware configuration
Custom	N/A	6	For user custom purposes. The size of this packet is user application dependent.

Note:

1. For details, refer to the section “Boot Strategies”.

The address of the Boot Configuration and Secure Boot Configuration special packets can be retrieved in the OTPC Boot Addresses (OTPC_BAR) register.

The address of the Custom special packets can be retrieved in the OTPC Custom Address (OTPC_CAR) register for non-secure packets and in the OTPC Secure Custom Address (OTPC_SCAR) register for secure packets.

For each “special” packets, if there is more than one valid packet (of the same type), only the last packet will be considered (previous packets will be ignored). It is recommended to invalidate prior “special” packets to keep only one valid packet for each “special” packet type.

30.5.3.4 Init

After each reset, the OTPC parses the User area to check its content. The header of each packet will be read, depending on the header value some actions can be triggered:

- If the header is corrupted, the init sequence is interrupted and the OTPC_ISR.COERR bit is set.
- If the INVLD field of the header is 3, the OTPC ignores the packet and jumps to the next header.
- If the LOCK bit of the header is set, the payload is read and the checksum computed during the read is compared to the checksum saved in the header. If the checksums do not match, the OTPC_ISR.CKERR bit will be set.
- If the PACKET field of the header is BOOT_CONFIGURATION, the address of the packet will be stored in the OTPC_BAR.BCADDR field. Any previous value stored in the OTPC_BAR.BCADDR field is lost.
- If the PACKET field of the header is SECURE_BOOT_CONFIGURATION, the address of the packet will be stored in the OTPC_BAR.SBCADDR field. Any previous value stored in the OTPC_BAR.SBCADDR field is lost.
- If the PACKET field of the header is HARDWARE_CONFIGURATION, the payload is read and stored in the OTPC User Hardware Configuration (OTPC_UHCxR) registers (unless the checksums do not match if the packet is locked, in that case the reset value of the OTPC_UHCxR registers is stored). Any previous value stored in the OTPC_UHCxR registers is lost.
- If the PACKET field of the header is CUSTOM and the SECURE bit is 0, the address of the packet will be stored in the OTPC_CAR.CADDR field. Any previous value stored in the OTPC_CAR.CADDR field is lost.
- If the PACKET field of the header is CUSTOM and the SECURE bit is 1, the address of the packet will be stored in the OTPC_SCAR.SCADDR field. Any previous value stored in the OTPC_SCAR.SCADDR field is lost.

At the end of the init sequence, the value of the last HARDWARE_CONFIGURATION “special“ packet found is applied to the hardware.

30.5.3.5 Read Access

The User area can be read through the User interface at any time after a reset. Each packet is available through the OTPC_DR register. To trigger a packet read, follow the steps below:

1. The address of the header (or any address of the payload) must be written in OTPC_MR.ADDR.
2. OTPC_CR.READ must be set to ‘1’ (the OTPC_CR.KEY field value does not matter).
3. Wait for OTPC_ISR.EOR to be set to ‘1’ or for OTPC_SR.READ to be set to ‘0’, indicating that the whole packet has been transferred into temporary registers.
4. Read the header of the packet in the OTPC_HR register. To read each payload word, the address of the payload word must be written in OTPC_AR.DADDR. The payload word is then available in OTPC_DR. If OTPC_AR.INCRT is set to AFTER_READ, any read in the OTPC_DR increments the DADDR field.

   If INCRT is set to AFTER_WRITE, any write in the OTPC_DR increments the DADDR field.

The payload of a packet with PACKET set to KEY is read as ‘0’. The payload of a packet hidden since the last reset is read as ‘0’.

30.5.3.5.1 Transfer a Packet through the Host Key Bus

To transfer a packet from the OTP memory through the host key bus, follow the steps below:

1. The address of the header (or any address of the payload) must be written in OTPC_MR.ADDR. Only the packets with the packet type set to KEY are transferable on the host key bus.
2. The key bus destination must be written in OTPC_MR.KBDST.
3. Write 0x7167in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.KBSTART.

The end of the transfer is indicated by OTPC_ISR.EOKT=’1’ and/or OTPC_SR.MKBB=’0’.

If the type of the packet is not KEY, OTPC_ISR.KBERR is set.

To cancel a packet transfer, OTPC_CR.KEY must be set to 0x7167 and KBSTOP must be set.

30.5.3.5.2 Hiding a Packet

For security reasons, it is possible to hide a packet after having read it through the User Interface. Once hidden, any read to the payload of the packet returns ‘0’.

To unhide a packet, a reset is necessary.

Hiding a packet does not make it available through the Key Buses.

To hide a packet, follow the steps below:

1. Write the address value of the header of the packet to hide in OTPC_MR.ADDR.
2. Write 0x7167 in OTPC_CR.KEY and ‘1’ to HIDE.

30.5.3.6 Write (Program) Considerations

Each word of the User area can be written only once. It is possible to write a packet payload partially and/or update a packet payload already written.

The packet to write (either the header or the payload) may contain some bits already at ‘1’. In this case, a dummy packet can be used to write the packet in a different location. The OTPC may also be able to fix the ‘1’ already written if this bit also matches the packet to write. Thus before writing any new packet, it is necessary to proceed to a read at the last address.

30.5.3.6.1 ‘1’ in the Header

After the read, the header may contain one or more ‘1’s. If the ‘1’s match the ‘1’s to write in the header, the packet can be written.

If the ‘1’s do not all match the ‘1’s to write in the header, the packet cannot be written. It is mandatory to create (and then invalidate if necessary) a packet with a compatible header.

30.5.3.6.2 ‘1’ in the Payload

If the payload is written and then updated later, the ‘1’s already written must match the ‘1’s to write.

If the payload is written only once (no update later), the ‘1’s already written must all match either the ‘1’ or the ‘0’ to write (it cannot be a mix of ‘1’s and ‘0’s to write).

30.5.3.7 Write (Program) Access

The User area can be programmed at any time after a reset until OTPC_MR.WRDIS has been set.

30.5.3.7.1 Writing a New Packet from the User Interface

To write a new packet from the User Interface, follow the steps below:

1. Write OTPC_MR.NPCKT to ‘0’ if it is set at ‘1’.
2. Write OTPC_MR.ADDR to its maximum value.
3. Write a ‘1’ to OTPC_CR.READ and wait for the read completion (OTPC_ISR.EOR=‘1’ when the read is completed).
4. Check there is no bit already set to ‘1’ in the OTPC_HR and OTPC_DR. The check for the data registers can be replaced by reading OTPC_SR.ONEF (if ONEF is set, at least one bit of the data registers is set to ‘1’).
   If the header or the payload of the packet already contains a 1, the new packet may need to be adapted.
5. Write OTPC_MR.ADDR to ‘0’ and set NPCKT.
   Depending on the contents of the temporary registers, an automatic flush can be triggered. If an automatic flush is started, OTPC_SR.FLUSH is set and OTPC_ISR.EOF is raised at the completion of the flush. It is mandatory to wait for the end of the flush.
6. Write the header value in OTPC_HR. The value of PACKET must not be the same as KEY.
7. Set DADDR to ‘0’.
8. Write the first data in the OTPC_DR register. To update the 32-bit data later (using the packet update), the OTPC_DR register must be set to 0.
9. Increment the DADDR field and write the next data in the OTPC_DR. Repeat this operation until all the data has been written.
   Skip the increment of DADDR if INCRT is set to AFTER_WRITE.
10. Write USER_KEY in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.

Before the write operation in the OTP memory, the OTPC checks the consistency of the packet and that the packet does not overlap on any existing packet. In case of error, OTPC_ISR.WERR is set and the write operation is cancelled.

The end of the programming operation is indicated by OTPC_ISR.EOP=’1’ and/or OTPC_SR.PGM=’0’. At the end of the programming, the address of the header is available in OTPC_MR.ADDR and the OTPC_MR.NPCKT must be cleared.

The payload can be read back before programming. After read back, it is possible to update PACKET value to KEY before programming.

If the new written packet is the User Hardware Configuration special packet, its payload is ignored until the next reset or the next refresh. The OTPC_UHCxR registers will be updated after the reset or the refresh following programming.

30.5.3.7.2 Updating an Existing Packet from the User Interface

To update an existing packet from the User Interface, follow the steps below:

1. Write the address of the header of the packet to update in OTPC_MR.ADDR.
2. Start a read by setting OTPC_CR.READ and wait for the read completion indicated by OTPC_ISR.EOR.
3. Update the data using the OTPC_AR and OTPC_DR registers. Only the 32-bit data set to 0 can be updated, the non-zero 32-bit data must be left unchanged
4. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.

The end of the programming operation is indicated by OTPC_ISR.EOP='1' and/or OTPC_SR.PGM='0'.

If the updated packet is the User Hardware Configuration special packet, its new payload is ignored until the next reset. The OTPC_UHCxR registers will be updated after the reset or the refresh following programming.

30.5.3.7.3 Writing a Packet from the Client Key Bus

To write a packet from the client key bus interface (payload only), follow the steps below:

1. Write OTPC_MR.ADDR to ‘0’ and set NPCKT.
   Depending on the content of the temporary registers, an automatic flush can be triggered. If an automatic flush is started, OTPC_SR.FLUSH is set and OTPC_ISR.EOF is raised at the completion of the flush. It is mandatory to wait for the end of the flush.
2. Write the header value in the OTPC_HR register. The value of PACKET must be KEY.
3. Initiate a data transfer to the OTP memory through the TRNG host key bus.
4. Wait for the data transfer completion.
5. Check that no error happened during the key transfer (OTPC_ISR.KBERR must be cleared).
6. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.
7. Before the write operation in the OTP memory, the OTPC checks the consistency of the packet and that the packet does not overlap on any existing packet. In case of error, OTPC_ISR.WERR is set and the write operation is cancelled. The end of the programming operation is indicated by OTPC_ISR.EOP=1 and/or OTPC_SR.PGM=0.

If the PACKET field is changed before programming, the payload is erased (and lost).

30.5.3.7.4 Locking a Packet

To lock a packet, follow the steps below:

1. Write the address value of the header of the packet to lock in OTPC_MR.ADDR.
2. Start a read by setting OTPC_CR.READ and waiting for the read completion indicated by OTPC_ISR.EOR.
3. Write 0x7167 in the OTPC_CR.KEY field and ‘1‘ in OTPC_CR.CKSGEN.

The end of the lock operation is indicated by OTPC_ISR.EOL=’1’ and/or OTPC_SR.LOCK=’0’.

Generating the checksum locks the packet and modification is no longer possible.

30.5.3.7.5 Invalidating a Packet

To invalidate a packet, follow the steps below:

1. Write the address value of the header of the packet to invalidate in OTPC_MR.ADDR.
2. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.INVLD.

The end of the invalidation operation is indicated by OTPC_ISR.EOI= ‘1’ and/or OTPC_SR.INVLD=’0’.

If the invalidated packet is the User Hardware Configuration special packet, its old payload remains active until the next reset or the next refresh. The OTPC_UHCxR registers will be updated after the reset or the refresh following the invalidation operation.

30.5.3.8 Fixing Corruption

During a programming sequence, packet header corruption may occur. This corruption can be caused by a partial programming of the header. It is mandatory to fix any corruption prior to any usage of the Engineering Area or User Area.

During the start sequence, the OTPC stops parsing the OTP memory at the first header corruption detected. When OTPC_ISR.COERR is set, a corruption has been detected. The corrupted header can be read in OTPC_HR and its location can be read in OTPC_MR.ADDR.

A header is corrupted if at least one of the following statements matches:

- The ONE bit is cleared (it must be set to fix the corruption).
- The INVLD field is 3 and the PACKET field is 0 (PACKET must be set to a non-0 value to fix the corruption).
- The SIZE and PACKET fields are not consistent (for PACKET set to PRODUCT_UID, HARDWARE_CONFIGURATION or SECURITY_CONFIGURATION, the packet must be invalidated to fix the corruption).

To fix a corruption, start a a read procedure at the location of the corrupted header. The OTPC reads the payload according to the size provided in the header and reads one extra word of payload, which should match the next header.

The corrupted header must be fixed by writing any missing '1's or, if not possible, by extending its size if the supposed next header is 0, or by invalidating the packet.

A reset is required after fixing the corruption.

30.5.3.9 "Software" Protections

The User area can be protected against read accesses and/or modifications.

To enable read protection of the User data (OTPC_DR) and header (OTPC_HR) registers, OTPC_MR.RDDIS must be set. Clearing RDDIS allows read access again. When the OTPC_DR and OTPC_HR registers are read-protected, any read returns 0.

To enable write protection of the OTPC_DR registers, the WRDIS bit of OTPC_MR should be set. Clearing the WRDIS bit allows write access again.

To enable write protection of the User area, the write protection of the User data registers must be enabled.

The OTPC_MR can be locked until the next reset by setting the LOCK bit of OTPC_MR. Once locked, the current protection configuration of the OTPC_DR and OTPC_HR registers applies, it is then also impossible to update, program, invalidate, hide or read a packet (the OTPC_MR.ADDR field is then locked too preventing to select a packet).

30.5.3.10 "Hardware" Protections

The User area can be protected against read accesses and/or modifications.

To enable the different protections of the User area, the User Configuration special packet must be programmed. The packet is described in the OTPC_UHCxR registers.

As an example, to disable the JTAG interface for an indefinite period, the JTAGDIS, UHCINVDIS and UHCPGDIS fields of the User Hardware Configuration special packet must all be programmed to a non-zero value. Thus, it will be impossible to update or invalidate the User Hardware Configuration special packet.

⚠WARNING	"Hardware" protections are in effect for an indefinite period and cannot be cancelled.

30.5.4 OTP Emulation Mode

The OTPC features an Emulation mode. This Emulation mode can be used to test all the operations allowed by the controller on a backed-up memory instead of the real OTP memory.

When Emulation mode is enabled, the controller has the same behavior. The Emulation mode is enabled only when the OTP memory has not been previously programmed.

To enable/disable Emulation mode on the User area, follow the steps below:

1. Set OTPC_MR.EMUL to '1' (to enable) or to '0' (to disable).
2. Refresh the User area by writing a '1' to OTPC_CR.REFRESH and 0x7167 in OTPC_CR.KEY.
3. Wait for the refresh completion by polling OTPC_ISR.EORF.

The current running mode of the User area can be observed by reading OTPC_SR.EMUL. If EMUL is set to '1', Emulation mode is enabled; if it is set to '0', Emulation mode is disabled.

After a reset, Emulation mode is disabled.

30.5.5 Interrupts

An OTPC interrupt request can be triggered when one or several of the following bits are set in the OTPC Interrupt Status register (OTPC_ISR): End Of Programming (EOP), End Of Locking (EOL), End Of Invalidation (EOI), End Of Key Transfer (EOKT), Programming Error (PGERR), Locking Error (LKERR), Invalidation Error (IVERR), Write Error (WERR), End Of Read (EOR), End Of Flush (EOF), End Of Hide (EOH), End Of Refresh (EORF), Checksum Check Error (CKERR) or Key Invalid Error (KBERR).

The interrupt request is generated if the corresponding bit in the OTPC Interrupt Mask register (OTPC_IMR) is set. Bits in OTPC_IMR are set by writing a '1' to the corresponding bit in the OTPC Interrupt Enable register (OTPC_IER) and cleared by writing a '1' to the corresponding bit in the OTPC Interrupt Disable register (OTPC_IDR). The interrupt request remains active until the corresponding bit in OTPC_ISR is cleared.

Reading the OTPC_ISR clears all bits of the register.

30.5.6 Register Write Protection

To prevent any single software error from corrupting the OTPC behavior, certain registers in the address space can be write-protected by setting the Write Protection Configuration Enable (WPCFEN), Write Protection Interrupt Enable (WPITEN) and/or Write Protection Control Enable (WPCTEN) bit(s) in the Write Protection Mode Register (OTPC_WPMR).

If a write access to the protected registers is detected, the Write Protection Violation Status (WPVS) flag in the Write Protection Status Register (OTPC_WPSR) is set and the field Write Protection Violation Source (WPVSRC) indicates the register in which the write access has been attempted. An interrupt can be raised if the Security and/or Safety Event (SECE) interrupt is set in OTPC_IMR.

The WPVS flag is automatically reset by reading the OTPC_WPSR.

The following registers can be write-protected with the OTPC_WPMR.WPCFEN bit:

- OTPC Mode Register

The following registers can be write-protected with the OTPC_WPMR.WPITEN bit:

- OTPC Interrupt Enable Register
- OTPC Interrupt Disable Register

The following registers can be write-protected with the OTPC_WPMR.WPCTEN bit:

- OTPC Control Register

30.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	OTPC_CR	31:24	KEY[15:8]							
		23:16	KEY[7:0]							
		15:8	REFRESH						KBSTOP	KBSTART
		7:0	FLUSH	READ		HIDE		INVLD	CKSGEN	PGM
0x04	OTPC_MR	31:24	ADDR[15:8]							
		23:16	ADDR[7:0]							
		15:8	LOCK		KBDST[1:0]				WRDIS	RDDIS
		7:0	EMUL			NPCKT				UHCRRDIS
0x08	OTPC_AR	31:24								
		23:16								INCRT
		15:8								
		7:0	DADDR[7:0]							
0x0C	OTPC_SR	31:24								
		23:16								
		15:8							ONEF	HIDE
		7:0	FLUSH	READ	SKBB	MKBB	EMUL	INVLD	LOCK	PGM
0x10	OTPC_IER	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x14	OTPC_IDR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x18	OTPC_IMR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x1C	OTPC_ISR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x20	OTPC_HR	31:24	CHECKSUM[15:8]							
		23:16	CHECKSUM[7:0]							
		15:8	SIZE[7:0]							
		7:0	ONE	SECURE	INVLD[1:0]		LOCK	PACKET[2:0]		
0x24	OTPC_DR	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x28 ... 0x2F	Reserved									
0x30	OTPC_BAR	31:24	SBCADDR[15:8]							
		23:16	SBCADDR[7:0]							
		15:8	BCADDR[15:8]							
		7:0	BCADDR[7:0]							
0x34	OTPC_CAR	31:24								
		23:16								
		15:8	CADDR[15:8]							
		7:0	CADDR[7:0]							
0x38	OTPC_SCAR	31:24								
		23:16								
		15:8	SCADDR[15:8]							
		7:0	SCADDR[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C ... 0x4F	Reserved									
0x50	OTPC_UHC0R	31:24								
		23:16								
		15:8	SECDBG[7:0]							
		7:0	JTAGDIS[7:0]							
0x54	OTPC_UHC1R	31:24								
		23:16							URFDIS	CPGDIS
		15:8	CLKDIS	CINVDIS	SCPGDIS	SCLKDIS	SCINVDIS	SBCPGDIS	SBCLKDIS	SBCINVDIS
		7:0	BCPGDIS	BCLKDIS	BCINVDIS	UHCPGDIS	UHCLKDIS	UHCINVDIS	UPGDIS	URDDIS
0x58 ... 0x5F	Reserved									
0x60	OTPC_UID0R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x64	OTPC_UID1R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x68	OTPC_UID2R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x6C	OTPC_UID3R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x70 ... 0xE3	Reserved									
0xE4	OTPC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCTEN	WPITEN	WPCFEN
0xE8	OTPC_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

30.6.1 OTPC Control Register

Name: OTPC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN bit is cleared in the OTPC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	REFRESH						KBSTOP	KBSTART
Access	W						W	W
Reset	–						–	–

Bit	7	6	5	4	3	2	1	0
	FLUSH	READ		HIDE		INVLD	CKSGEN	PGM
Access	W	W		W		W	W	W
Reset	–	–		–		–	–	–

Bits 31:16 – KEY[15:0] Programming Key
This field must be written with the correct key code (0x7167) to allow programming, checksum generation, packet invalidation or packet hiding.

Bit 15 – REFRESH Refresh the Area

Value	Description
0	No effect.
1	Starts a refresh of the area.

Bit 9 – KBSTOP Key Bus Transfer Stop

Value	Description
0	No effect.
1	Stops an on-going transfer on the host key bus.

Bit 8 – KBSTART Key Bus Transfer Start

Value	Description
0	No effect.
1	Starts a transfer through the host key bus.

Bit 7 – FLUSH Flush Temporary Registers

Value	Description
0	No effect.
1	Starts a flush of the temporary registers used to store the packet payload.

Bit 6 – READ Read Packet

Value	Description
0	No effect.
1	Starts a read sequence of the selected packet.

Bit 4 – HIDE Hide Packet

Value	Description
0	No effect.
1	The selected packet is not readable anymore until the next reset.

Bit 2 – INVLD Invalidate Packet

Value	Description
0	No effect.
1	Invalidates the selected packet.

Bit 1 – CKSGEN Generate Checksum

Value	Description
0	No effect.
1	Generates and programs the selected packet checksum. This action also locks the packet.

Bit 0 – PGM Program Packet

Value	Description
0	No effect.
1	The selected packet is written.

30.6.2 OTPC Mode Register

Name: OTPC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in the OTPC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LOCK		KBDST[1:0]				WRDIS	RDDIS
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bit	7	6	5	4	3	2	1	0
	EMUL			NPCKT				UHCRRDIS
Access	R/W			R/W				R/W
Reset	0			0				0

Bits 31:16 – ADDR[15:0] Address
This field represents the address of the packet's header.

Bit 15 – LOCK Lock Register
The LOCK bit is set-only. Only a reset can disable lock.

Value	Description
0	The OTPC_MR register is unlocked; write access changes its value.
1	The OTPC_MR register is locked; write access does not change its value.

Bits 13:12 – KBDST[1:0] Key Bus Destination

Value	Name	Description
0	NONE	No destination selected (no transfer can occur).
1	AES	The AES is the destination of the key transfer.
2	TZAESB	The TrustZone AES Bridge is the destination of the key transfer.
3	TDES	The TDES is the destination of the key transfer.

Bit 9 – WRDIS Write Disable

Value	Description
0	The write capability of the OTPC_DR register is enabled.
1	The write capability of the OTPC_DR register is disabled.

Bit 8 – RDDIS Read Disable

Value	Description
0	The read capability of the OTPC_HR and OTPC_DR registers are enabled.
1	The read capability of the OTPC_HR and OTPC_DR registers are disabled. In case of read, the returned value is 0.

Bit 7 – EMUL Emulation Enable

Value	Description
0	The Emulation mode of the User area is disabled, all accesses are computed in the OTP memory.
1	The Emulation mode of the User area is enabled, all accesses are computed in the Emulation memory.

Bit 4 – NPCKT New Packet

Value	Description
0	Updates the packet defined at the ADDR address.
1	Creates a new packet.

Bit 0 – UHCRRDIS User Hardware Configuration Register Read Disable

Value	Description
0	The User Hardware Configuration register can be read through the User Interface.
1	The User Hardware Configuration register cannot be read through the User Interface.

30.6.3 OTPC Address Register

Name: OTPC_AR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								INCRT
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – INCRT Increment Type

Value	Name	Description
0	AFTER_READ	Increment DADDR after a read of OTPC_DR.
1	AFTER_WRITE	Increment DADDR after a write of OTPC_DR.

Bits 7:0 – DADDR[7:0] Data Address
This field represents the word address of the payload to access through the OTPC_DR register.

30.6.4 OTPC Status Register

Name: OTPC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							ONEF	HIDE
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	FLUSH	READ	SKBB	MKBB	EMUL	INVLD	LOCK	PGM
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – ONEF One Found

Value	Description
0	No bit at ‘1’ found during the last packet read.
1	At least one ‘1’ has been found during the last packet read.

Bit 8 – HIDE Hiding On-Going

Value	Description
0	No packet hiding is on-going.
1	A packet hiding is on-going.

Bit 7 – FLUSH Flush On-Going

Value	Description
0	The temporary registers are not flushed.
1	The temporary registers are being flushed.

Bit 6 – READ Read On-Going

Value	Description
0	No packet read is on-going.
1	A packet read is on-going.

Bit 5 – SKBB Client Key Bus Busy

Value	Description
0	The client key bus is not busy.
1	The client key bus is busy.

Bit 4 – MKBB Host Key Bus Busy

Value	Description
0	The host key bus is not busy.

Value	Description
1	The host key bus is busy.

Bit 3 – EMUL Emulation Enabled

Value	Description
0	The User area Emulation mode is disabled.
1	The User area Emulation mode is enabled.

Bit 2 – INVLD Invalidation On-Going

Value	Description
0	No packet invalidation is on-going.
1	A packet invalidation is on-going.

Bit 1 – LOCK Lock On-Going

Value	Description
0	No packet locking is on-going.
1	A packet locking is on-going.

Bit 0 – PGM Programming On-Going

Value	Description
0	No packet programming is on-going.
1	A packet programming is on-going.

30.6.5 OTPC Interrupt Enable Register

Name: OTPC_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the OTPC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – SECE Security and/or Safety Event Interrupt Enable

Bit 16 – KBERR Key Bus Error Interrupt Enable

Bit 14 – HDERR Hide Error Interrupt Enable

Bit 13 – COERR Corruption Error Interrupt Enable

Bit 12 – CKERR Checksum Check Error Interrupt Enable

Bit 11 – EORF End Of Refresh Interrupt Enable

Bit 10 – EOH End Of Hide Interrupt Enable

Bit 9 – EOF End Of Flush Interrupt Enable

Bit 8 – EOR End Of Read Interrupt Enable

Bit 7 – WERR Write Error Interrupt Enable

Bit 6 – IVERR Invalidation Error Interrupt Enable

Bit 5 – LKERR Locking Error Interrupt Enable

Bit 4 – PGERR Programming Error Interrupt Enable

Bit 3 – EOKT End Of Key Transfer Interrupt Enable

Bit 2 – EOI End Of Invalidation Interrupt Enable

Bit 1 – EOL End Of Locking Interrupt Enable

Bit 0 – EOP End Of Programming Interrupt Enable

30.6.6 OTPC Interrupt Disable Register

Name: OTPC_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the OTPC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – SECE Security and/or Safety Event Interrupt Disable

Bit 16 – KBERR Key Bus Error Interrupt Disable

Bit 14 – HDERR Hide Error Interrupt Disable

Bit 13 – COERR Corruption Error Interrupt Disable

Bit 12 – CKERR Checksum Check Error Interrupt Disable

Bit 11 – EORF End Of Refresh Interrupt Disable

Bit 10 – EOH End Of Hide Interrupt Disable

Bit 9 – EOF End Of Flush Interrupt Disable

Bit 8 – EOR End Of Read Interrupt Disable

Bit 7 – WERR Write Error Interrupt Disable

Bit 6 – IVERR Invalidation Error Interrupt Disable

Bit 5 – LKERR Locking Error Interrupt Disable

Bit 4 – PGERR Programming Error Interrupt Disable

Bit 3 – EOKT End Of Key Transfer Interrupt Disable

Bit 2 – EOI End Of Invalidation Interrupt Disable

Bit 1 – EOL End Of Locking Interrupt Disable

Bit 0 – EOP End Of Programming Interrupt Disable

30.6.7 OTPC Interrupt Mask Register

Name: OTPC_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: Corresponding interrupt is not enabled.

1: Corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 28 – SECE Security and/or Safety Event Interrupt Mask

Bit 16 – KBERR Key Bus Error Interrupt Mask

Bit 14 – HDERR Hide Error Interrupt Mask

Bit 13 – COERR Corruption Error Interrupt Mask

Bit 12 – CKERR Checksum Check Error Interrupt Mask

Bit 11 – EORF End Of Refresh Interrupt Mask

Bit 10 – EOH End Of Hide Interrupt Mask

Bit 9 – EOF End Of Flush Interrupt Mask

Bit 8 – EOR End Of Read Interrupt Mask

Bit 7 – WERR Write Error Interrupt Mask

Bit 6 – IVERR Invalidation Error Interrupt Mask

Bit 5 – LKERR Locking Error Interrupt Mask

Bit 4 – PGERR Programming Error Interrupt Mask

Bit 3 – EOKT End Of Key Transfer Interrupt Mask

Bit 2 – EOI End Of Invalidation Interrupt Mask

Bit 1 – EOL End Of Locking Interrupt Mask

Bit 0 – EOP End Of Programming Interrupt Mask

30.6.8 OTPC Interrupt Status Register

Name: OTPC_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 28 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred since the last read of OTPC_ISR.
1	One or more safety or security event occurred since the last read of OTPC_ISR. For details on the event, refer to OTPC_WPSR.

Bit 16 – KBERR Key Bus Error (cleared on read)

Value	Description
0	No error happened on the key bus since the last read of OTPC_ISR.
1	An error happened on the key bus since the last read of OTPC_ISR.

Bit 14 – HDERR Hide Error (cleared on read)

Value	Description
0	No hiding error occurred since the last read of OTPC_ISR.
1	A hiding error occurred since the last read of OTPC_ISR.

Bit 13 – COERR Corruption Error (cleared on read)

Value	Description
0	No corruption occurred during the last start-up since the last read of OTPC_ISR.
1	A corruption occurred since the last read of OTPC_ISR.

Bit 12 – CKERR Checksum Check Error (cleared on read)

Value	Description
0	No checksum check failure occurred during last reading sequence since the last read of OTPC_ISR.
1	A checksum check failure occurred since the last read of OTPC_ISR.

Bit 11 – EORF End Of Refresh (cleared on read)

Value	Description
0	No refresh sequence completion since the last read of OTPC_ISR.
1	At least one refresh sequence completion since the last read of OTPC_ISR.

Bit 10 – EOH End Of Hide (cleared on read)

Value	Description
0	No hiding sequence completion since the last read of OTPC_ISR.
1	At least one hiding sequence completion since the last read of OTPC_ISR.

Bit 9 – EOF End Of Flush (cleared on read)

Value	Description
0	No flush of the temporary registers since the last read of OTPC_ISR.
1	At least one flush hof the temporary registers has been completed since the last read of OTPC_ISR.

Bit 8 – EOR End Of Read (cleared on read)

Value	Description
0	No reading sequence completion since the last read of OTPC_ISR.
1	At least one reading sequence completion since the last read of OTPC_ISR.

Bit 7 – WERR Write Error (cleared on read)

Value	Description
0	No write error occurred since the last read of OTPC_ISR.
1	A write error occurred since the last read of OTPC_ISR.

Bit 6 – IVERR Invalidation Error (cleared on read)

Value	Description
0	No invalidation failure occurred during last invalidation sequence since the last read of OTPC_ISR.
1	An invalidation failure occurred since the last read of OTPC_ISR.

Bit 5 – LKERR Locking Error (cleared on read)

Value	Description
0	No locking failure occurred during last locking sequence since the last read of OTPC_ISR.
1	A locking failure occurred since the last read of OTPC_ISR.

Bit 4 – PGERR Programming Error (cleared on read)

Value	Description
0	No programming failure occurred during last programming sequence since the last read of OTPC_ISR.
1	A programming failure occurred since the last read of OTPC_ISR.

Bit 3 – EOKT End Of Key Transfer (cleared on read)

Value	Description
0	No key transfer completion since the last read of OTPC_ISR.
1	At least one key transfer has been completed on the host key bus since the last read of OTPC_ISR.

Bit 2 – EOI End Of Invalidation (cleared on read)

Value	Description
0	No invalidation sequence completion since the last read of OTPC_ISR.
1	At least one invalidation sequence completion since the last read of OTPC_ISR.

Bit 1 – EOL End Of Locking (cleared on read)

Value	Description
0	No locking sequence completion since the last read of OTPC_ISR.
1	At least one locking sequence completion since the last read of OTPC_ISR.

Bit 0 – EOP End Of Programming (cleared on read)

Value	Description
0	No programming sequence completion since the last read of OTPC_ISR.
1	At least one programming sequence completion since the last read of OTPC_ISR.

30.6.9 OTPC Header Register

Name: OTPC_HR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHECKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHECKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONE	SECURE	INVLD[1:0]		LOCK	PACKET[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – CHECKSUM[15:0] Packet Checksum
When CHECKSUM is read as 0, the checksum has not been generated. It is still possible to modify the packet content.
When CHECKSUM is read as 0x33CC, the checksum has not been generated but has already some bit at 1. Locking the packet may fail.
When CHECKSUM is read as 0xA5A5, the checksum has been generated and the last check was successful. It is impossible to modify the packet content.
When CHECKSUM is read as 0xCC33, the packet header is corrupted.
When CHECKSUM is read as 0xFFFF, the entire packet is no longer valid. It is possible to modify the packet content and it is up to the software to program the payload to a different value if needed.
For all other values of CHECKSUM, the checksum has been generated and the last check failed to match the checksum written in the OTP. It is impossible to modify the packet content.
This field is not writeable and is set by the OTPC during a lock request.

Bits 15:8 – SIZE[7:0] Packet Size
This field represents the size of the packet payload.
This field is writeable only for new packets.

Bit 7 – ONE One
This field is set to 1 by hardware and is not writeable.

Bit 6 – SECURE Secure Packet
This field is writeable only through accesses done in the secure world, otherwise it is set to 0.

Value	Description
0	The packet is not part of the secure world.
1	The packet is part of the secure world.

Bits 5:4 – INVLD[1:0] Invalid Status
If set to value 3, this field indicates that the packet is not valid.

This field is not writeable and is set by the OTPC during an invalidation request.

Bit 3 – LOCK Lock Status
This field is not writeable and is set by the OTPC during a lock request.

Value	Description
0	The packet is not locked.
1	The packet is locked.

Bits 2:0 – PACKET[2:0] Packet Type
This field is writeable only for new packets.

Value	Name	Description
1	REGULAR	Regular packet accessible through the user interface
2	KEY	Key packet accessible only through the Key Buses
3	BOOT_CONFIGURATION	Boot Configuration packet
4	SECURE_BOOT_CONFIGURATION	Secure Boot Configuration packet
5	HARDWARE_CONFIGURATION	Hardware Configuration packet
6	CUSTOM	Custom packet

30.6.10 OTPC Data Register

Name: OTPC_DR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Packet Data
This field represents the data of one of the packets. The data read or written is located at the address specified by OTPC_AR.DADDR.

30.6.11 OTPC Boot Addresses Register

Name: OTPC_BAR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SBCADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SBCADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BCADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BCADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – SBCADDR[15:0] Secure Boot Configuration Address
This field represents the address of the “Secure Boot Configuration” special packet.

Bits 15:0 – BCADDR[15:0] Boot Configuration Address
This field represents the address of the “Boot Configuration” special packet.

30.6.12 OTPC Custom Address Register

Name: OTPC_CAR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CADDR[15:0] Custom Address
This field represents the address of the “Custom” special packet (Non-Secure packet).

30.6.13 OTPC Secure Custom Address Register

Name: OTPC_SCAR
Offset: 0x38
Reset: 0x00000000
Property: Read-only

This register can only be read in the Secure world.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	SCADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SCADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SCADDR[15:0] Secure Custom Address
This field represents the address of the “Custom” special packet (Secure packet).

30.6.14 OTPC User Hardware Configuration 0 Register

Name: OTPC_UHC0R
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SECDBG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	JTAGDIS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – SECDBG[7:0] Secure Debug

Value	Description
0	The secure debug is allowed.
Non-Zero	The secure debug is forbidden.

Bits 7:0 – JTAGDIS[7:0] JTAG Disable

Value	Description
0	The JTAG is enabled.
Non-zero	The JTAG is disabled.

30.6.15 OTPC User Hardware Configuration 1 Register

Name: OTPC_UHC1R
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							URFDIS	CPGDIS
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLKDIS	CINVDIS	SCPGDIS	SCLKDIS	SCINVDIS	SBCPGDIS	SBCLKDIS	SBCINVDIS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCPGDIS	BCLKDIS	BCINVDIS	UHCPGDIS	UHCLKDIS	UHCINVDIS	UPGDIS	URDDIS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 17 – URFDIS User Refresh Disable

Value	Description
0	The OTPC_CR.REFRESH bit is fully functional.
1	The OTPC_CR.REFRESH bit is only functional in Emulation mode.

Bit 16 – CPGDIS Custom Packet Program Disable

Value	Description
0	The programming of Custom Special Packet is allowed.
1	The programming of Custom Special Packet is forbidden.

Bit 15 – CLKDIS Custom Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Custom Special Packet is allowed.
1	The generation of the checksum (lock) of the Custom Special Packet is forbidden.

Bit 14 – CINVDIS Custom Packet Invalidation Disable

Value	Description
0	The invalidation of the Custom Special Packet is allowed.
1	The invalidation of the Custom Special Packet is forbidden.

Bit 13 – SCPGDIS Secure Custom Packet Program Disable

Value	Description
0	The programming of Secure Custom Special Packet is allowed.
1	The programming of Secure Custom Special Packet is forbidden.

Bit 12 – SCLKDIS Secure Custom Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Secure Custom Special Packet is allowed.

Value	Description
1	The generation of the checksum (lock) of the Secure Custom Special Packet is forbidden.

Bit 11 – SCINVDIS Secure Custom Packet Invalidation Disable

Value	Description
0	The invalidation of the Secure Custom Special Packet is allowed.
1	The invalidation of the Secure Custom Special Packet is forbidden.

Bit 10 – SBCPGDIS Secure Boot Configuration Packet Program Disable

Value	Description
0	The programming of Secure Boot Configuration Special Packet is allowed.
1	The programming of Secure Boot Configuration Special Packet is forbidden.

Bit 9 – SBCLKDIS Secure Boot Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Secure Boot Configuration Special Packet is allowed.
1	The generation of the checksum (lock) of the Secure Boot Configuration Special Packet is forbidden.

Bit 8 – SBCINVDIS Secure Boot Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the Secure Boot Configuration Special Packet is allowed.
1	The invalidation of the Secure Boot Configuration Special Packet is forbidden.

Bit 7 – BCPGDIS Boot Configuration Packet Program Disable

Value	Description
0	The programming of Boot Configuration Special Packet is allowed.
1	The programming of Boot Configuration Special Packet is forbidden.

Bit 6 – BCLKDIS Boot Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Boot Configuration Special Packet is allowed.
1	The generation of the checksum (lock) of the Boot Configuration Special Packet is forbidden.

Bit 5 – BCINVDIS Boot Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the Boot Configuration Special Packet is allowed.
1	The invalidation of the Boot Configuration Special Packet is forbidden.

Bit 4 – UHCPGDIS User Hardware Configuration Packet Program Disable

Value	Description
0	The programming of User Hardware Configuration Special Packet is allowed.
1	The programming of User Hardware Configuration Special Packet is forbidden.

Bit 3 – UHCLKDIS User Hardware Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the User Hardware Configuration Special Packet is allowed.
1	The generation of the checksum (lock) of the User Hardware Configuration Special Packet is forbidden.

Bit 2 – UHCINVDIS User Hardware Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the User Hardware Configuration Special Packet is allowed.
1	The invalidation of the User Hardware Configuration Special Packet is forbidden.

Bit 1 – UPGDIS User programming Disable

Value	Description
0	The OTPC_CR.PGM bit is fully functional.

Value	Description
1	The OTPC_CR.PGM bit is not functional.

Bit 0 – URDDIS User Read Disable

Value	Description
0	The OTPC_CR.READ bit is fully functional.
1	The OTPC_CR.READ bit is not functional.

30.6.16 OTPC Product UID x Register

Name: OTPC_UIDxR
Offset: 0x60 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
	UID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UID[31:0] Unique Product ID
This field represents the unique product ID.

30.6.17 OTPC Write Protection Mode Register

Name: OTPC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCTEN	WPITEN	WPCFEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4F5450	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in OTPC_WPSR.WPVSRC and the last software control error type is reported in OTPC_WPSR.SWETYP; The OTPC_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in OTPC_WPSR.WPVSRC and only the first software control error type is reported in OTPC_WPSR.SWETYP. The OTPC_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCTEN Write Protection Control Enable

Value	Description
0	Disables the write protection of the control if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the control if WPKEY matches to 0x4F5450 (OTP in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection of the interruption configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the interruption configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).

Bit 0 – WPCFEN Write Protection Configuration Enable

Value	Description
0	Disables the write protection of the configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).

30.6.18 OTPC Write Protection Status Register

Name: OTPC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class

Value	Name	Description
0	WARNING	An abnormal access that does not have any impact.
1	ERROR	An abnormal access that may have an impact.

Bits 27:24 – SWETYP[3:0] Software Error Type

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	CONF_CHG	A change has been made into the configuration (error).
3	KEY_ERROR	A write has been computed in OTPC_CR or OTPC_WPMR register with a wrong value in the related KEY field (error).
4	DATA_ACC	The non-secure world application tried to read a packet from the secure world (error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of OTPC_WPSR.
1	A software error has occurred since the last read of OTPC_WPSR. The field SWETYP details the type of software error encountered.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of OTPC_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of OTPC_WPSR. This flag can be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of OTPC_WPSR.
1	A clock glitch has occurred since the last read of OTPC_WPSR. This flag can be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of OTPC_WPSR.
1	A write protection violation has occurred since the last read of OTPC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into WPVSRC.

31. Special Function Registers (SFR)

31.1 Description

Special Function Registers (SFR) manage specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

31.2 Embedded Characteristics

- 32-bit Special Function Registers Control Specific Behavior of the Product

31.3 Functional Description

31.3.1 Register Write Protection

To prevent any single software error from corrupting SFR behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the SFR Write Protection Mode Register (SFR_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SFR Write Protection Status Register (SFR_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted. To read this flag, the SFR peripheral clock must be enabled.

The WPVS bit is automatically cleared after reading the SFR_WPSR.

All registers can be write-protected by setting SFR_WPMR.WPEN.

31.4 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SFR_OHCIICR	31:24								
		23:16								
		15:8						SUSPEND2	SUSPEND1	SUSPEND0
		7:0							APPSTART	ARIE
0x04	SFR_OHCIISR	31:24								
		23:16								
		15:8								
		7:0						RIS2	RIS1	RIS0
0x08 ... 0xE3	Reserved									
0xE4	SFR_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SFR_WPSR	31:24								
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0								WPVS
0xEC ... 0x201F	Reserved									
0x2020	SFR_EHCIOHCI	31:24								
		23:16								
		15:8								
		7:0								EHCICLK
0x2024 ... 0x2027	Reserved									
0x2028	SFR_HSS_AXIQOS	31:24								
		23:16								
		15:8					WRITE[3:0]			
		7:0					READ[3:0]			
0x202C	SFR_UDDRC	31:24								
		23:16								
		15:8								
		7:0								DIS_DECERR
0x2030	SFR_CAN_SRAM_SEL	31:24								
		23:16								
		15:8								
		7:0			UPPER_CAN5	UPPER_CAN4	UPPER_CAN3	UPPER_CAN2	UPPER_CAN1	UPPER_CAN0
0x2034 ... 0x203F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2040	SFR_UTMI0R0	31:24							VBUS	TXPREEMPAMPTUNE[1]
		23:16	TXPREEMPAMPTUNE[0]							
		15:8								
		7:0					COMMONONN			
0x2044	SFR_UTMI0R1	31:24							VBUS	TXPREEMPAMPTUNE[1]
		23:16	TXPREEMPAMPTUNE[0]							
		15:8								
		7:0					COMMONONN			
0x2048	SFR_UTMI0R2	31:24							VBUS	TXPREEMPAMPTUNE[1]
		23:16	TXPREEMPAMPTUNE[0]							
		15:8								
		7:0					COMMONONN			

31.4.1 OHCI Interrupt Configuration Register

Name: SFR_OHCIICR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SUSPEND2	SUSPEND1	SUSPEND0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
							APPSTART	ARIE
Access							R/W	R/W
Reset							0	0

Bits 8, 9, 10 – SUSPENDx USB PORT x

Value	Description
0	Suspends controlled by EHCI-OCHI.
1	Forces the suspend for PORTx.

Bit 1 – APPSTART Reserved
Must write ‘0’.

Bit 0 – ARIE OHCI Asynchronous Resume Interrupt Enable

Value	Description
0	Interrupt is disabled.
1	Interrupt is enabled.

31.4.2 OHCI Interrupt Status Register

Name: SFR_OHCIISR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RIS2	RIS1	RIS0
Access						R	R	R
Reset						0	0	0

Bits 0, 1, 2 – RISx OHCI Resume Interrupt Status Port x

Value	Description
0	OHCI port resume not detected.
1	OHCI port resume detected.

31.4.3 SFR Write Protection Mode Register

Name: SFR_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534652	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
All registers are write-protected.

Value	Description
0	Disables the write protection if WPKEY value corresponds to 0x534652 (“SFR” in ASCII).
1	Enables the write protection if WPKEY value corresponds to 0x534652 (“SFR” in ASCII).

31.4.4 SFR Write Protection Status Register

Name: SFR_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

All registers can be write-protected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of SFR_WPSR.
1	A write protection violation has occurred since the last read of SFR_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

31.4.5 EHCIOHCI Register

Name: SFR_EHCIOHCI
Offset: 0x2020
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								EHCICLK
Access								R/W
Reset								0

Bit 0 – EHCICLK EHCI Clock Mode

Value	Description
0	FREECLK of Port A is used to drive the EHCI clock.
1	PHYCLK of Port A is used to drive the EHCI clock.

31.4.6 HSS AXI QOS Register

Name: SFR_HSS_AXIQOS
Offset: 0x2028
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					WRITE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					READ[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 11:8 – WRITE[3:0] Defines the QOS value for the write transfer from HSS to DDR controller

Bits 3:0 – READ[3:0] Defines the QOS value for the read transfer from HSS to DDR controller

31.4.7 UDDRC Register

Name: SFR_UDDRC
Offset: 0x202C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DIS_DECERR
Access								R/W
Reset								0

Bit 0 – DIS_DECERR Disable Decode Error

Value	Description
0	Enables decode error when reading a non-existing register in the DDR controller Configuration Address Space.
1	Disables decode error when reading a non-existing register in the DDR controller Configuration Address Space.

31.4.8 SFR CAN SRAM Selection Register

Name: SFR_CAN_SRAM_SEL
Offset: 0x2030
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			UPPER_CAN5	UPPER_CAN4	UPPER_CAN3	UPPER_CAN2	UPPER_CAN1	UPPER_CAN0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5 – UPPER_CANx CANx Upper 64K SRAM Selection

Value	Description
0	CANx accesses the lower 64K SRAM.
1	CANx accesses the upper 64K SRAM.

31.4.9 UTMI0 Configuration Register

Name: SFR_UTMI0Rx
Offset: 0x2040 + x*0x04 [x=0..2]
Reset: 0x029AD818 (SFR_UTMI0R0) or 0x009AD818 (SFR_UTMI0R1/R2)
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							VBUS	TXPREEMPAMPTUNE[1]
Access							R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	TXPREEMPAMPTUNE[0]							
Access	R/W							
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					COMMONONN			
Access					R/W			
Reset								

Bit 25 – VBUS VBUS Valid Indicator

Value	Description
0	The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.
1	The VBUS signal is valid, and the pull-up resistor on D+ is enabled.

Bits 24:23 – TXPREEMPAMPTUNE[1:0] HS Transmitter Pre-Emphasis Current Control
Controls the amount of current sourced to HSS#_DP and HSS#_DM after a J-to-K or K-to-J transition. The HS Transmitter preemphasis current is defined in terms of unit amounts. One unit amount is approximately 600 μA and is defined as 1X pre-emphasis current.

Value	Name
00	HS Transmitter pre-emphasis is disabled (default).
01	HS Transmitter pre-emphasis circuit sources 1x pre-emphasis current (recommended).
10	HS Transmitter pre-emphasis circuit sources 2x pre-emphasis current.
11	HS Transmitter pre-emphasis circuit sources 3x pre-emphasis current.

Bit 3 – COMMONONN Common Block Power-Down Control

Value	Description
0	PLL blocks remain powered.
1	PLL blocks are powered down.

32. Special Function Registers Backup (SFRBU)

32.1 Description

Special Function Registers Backup (SFRBU) manages specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

32.2 Embedded Characteristics

- 32-bit Special Function Registers Backup Controls Specific Behavior of the Product

32.3 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SFRBU_PSWBU	31:24	PSWKEY[23:16]							
		23:16	PSWKEY[15:8]							
		15:8	PSWKEY[7:0]							
		7:0						STATE	SOFTSWITCH	CTRL
0x04 ... 0x0B	Reserved									
0x0C	SFRBU_LDO25CR	31:24	LDOANAKEY[23:16]							
		23:16	LDOANAKEY[15:8]							
		15:8	LDOANAKEY[7:0]							
		7:0					STATE	LOWPOWER	PD_VALUE[1:0]	
0x10	SFRBU_DDRPWR	31:24								
		23:16								
		15:8								
		7:0								RETENTION

32.3.1 SFRBU Power Switch BU Control Register

Name: SFRBU_PSWBU
Offset: 0x00
Reset: 0x00000001
Property: Read/Write

The backup power switch selects the primary power source for the backup regulator (either VBAT or VDDIN33).

Bit	31	30	29	28	27	26	25	24
	PSWKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PSWKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PSWKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						STATE	SOFTSWITCH	CTRL
Access						R	R/W	R/W
Reset						0	0	1

Bits 31:8 – PSWKEY[23:0] Specific Value Mandatory to Allow Writing of Other Register Bits (Write-only)
PSWKEY is a security key preventing power switch changes due to software error or malicious code.

Value	Name	Description
0x4BD20C	PASSWD	Writing any other value in this field aborts the write operation in the SFRBU_PSWBU register. Always reads as 0.

Bit 2 – STATE Power Switch BU State (Read-only)
Reflects the power switch BU supply source selection in real time. After a switching request, the user must wait for the analog cell switching time to have an updated status (see the section Electrical Characteristics).

Value	Name	Description
0	VBAT	LDO Supply source is VBAT.
1	VDDIN33	LDO Supply source is VDDIN33.

Bit 1 – SOFTSWITCH Power Switch BU Source Selection
Has an action only if CTRL is set to ‘1’.

Value	Name	Description
0	VBAT	LDO Supply source is VBAT.
1	VDDIN33	LDO Supply source is VDDIN33.

Bit 0 – CTRL Power Switch BU Control
Used to control the Power Switch BU state by software.

Value	Name	Description
0	HARD	Power Switch BU is controlled by hardware (SOFTSWITCH bit has no action).
1	SOFT	Power Switch BU is controlled by software (SOFTSWITCH bit has an action).

32.3.2 SFRBU 2.5V LDO Control Register

Name: SFRBU_LDO25CR
Offset: 0x0C
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	LDOANAKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	LDOANAKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	LDOANAKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					STATE	LOWPOWER	PD_VALUE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bits 31:8 – LDOANAKEY[23:0] Specific value mandatory to allow writing of other register bits.
LDOANAKEY is a security key preventing power switch changes due to software error or malicious code.

Value	Name	Description
0x3B6E18	PASSWD	Writing any other value in this field aborts the write operation in the SFRBU_LDO25CR register. Always reads as 0.

Bit 3 – STATE LDOANA Switch On/Off Control

Value	Name	Description
0	ON	Switches LDOANA on.
1	OFF	Switches LDOANA off.

Bit 2 – LOWPOWER LDOANA Low-Power Mode Control

Value	Name	Description
0	NOT_SET	Switches LDOANA Low-power mode off.
1	SET	Switches LDOANA Low-power mode on.

Bits 1:0 – PD_VALUE[1:0] LDOANA Pull-down Value
LDOANA pull-down value plugged on LDO output when off.

Value	Name	Description
0	NONE	No pull-down plugged on LDO output when off.
1	500_OHMS	500 Ohms pull-down plugged on LDO output when off.
2	200_OHMS	200 Ohms pull-down plugged on LDO output when off.
3	100_OHMS	100 Ohms pull-down plugged on LDO output when off.

32.3.3 SFRBU DDR Power Control Register

Name: SFRBU_DDRPWR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								RETENTION
Access								R/W
Reset								0

Bit 0 – RETENTION SDRAM I/Os Retention
When set, SDRAM I/Os are in Retention state. Refer to “Backup with SDRAM in Self-Refresh (BSR) Mode” in the section “Electrical Characteristics” for more details.

33. Slow Clock Controller (SCKC)

33.1 Description

The System Controller embeds a Slow Clock Controller (SCKC). The SCKC selects the slow clock for the RTT and the RTC from one of two sources:

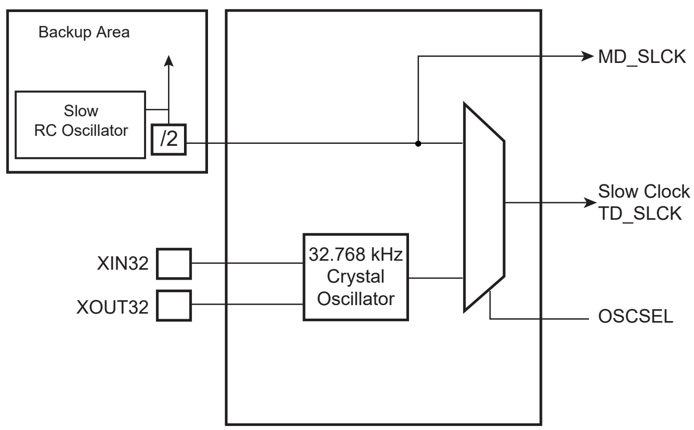
- External 32.768 kHz crystal oscillator
- Embedded always-on 64 kHz (typical) slow RC oscillator

33.2 Embedded Characteristics

- Always-on 64 kHz (Typical) Slow RC Oscillator or 32.768 kHz Crystal Oscillator Selector
- VDDBU-Powered

33.3 Block Diagram

Figure 33-1. SCKC Block Diagram



33.4 Functional Description

The TD_OSCSEL bit located in the Slow Clock Controller Configuration register (SCKC_CR) is in the backup domain and its value is kept while VBAT is present.

The embedded always-on 64 kHz (typical) slow RC oscillator is always enabled as soon as VBAT is established. The Slow Clock Selector command TD_OSCSEL bit selects the slow clock source of the RTT and the RTC.

After the VBAT Power-On-Reset, the default configuration is TD_OSCSEL = 0.

The programmer controls the slow clock switching by software, so precautions must be taken during the switching phase.

33.4.1 Switching from Embedded Always-on 64 kHz RC Oscillator to 32.768 kHz Crystal Oscillator

The sequence to switch from the embedded always-on 64 kHz (typical) slow RC oscillator to the 32.768 kHz crystal oscillator is the following:

1. Switch the main system bus clock to a source different from slow clock (PLL or Main Oscillator) through the Power Management Controller.
2. Switch from the embedded always-on 64 kHz RC oscillator to the 32.768 kHz crystal oscillator by writing a 1 to the TD_OSCSEL bit.

3. Wait 5 slow clock cycles for internal resynchronization.

33.4.2 Switching from 32.768 kHz Crystal Oscillator to Embedded Always-on 64 kHz RC Oscillator

The sequence to switch from the 32.768 kHz crystal oscillator to the embedded always-on 64 kHz (typical) RC oscillator is the following:

1. Switch the main system bus clock to a source different from slow clock (PLL or Main Oscillator).
2. Switch from the 32.768 kHz crystal oscillator to the embedded RC oscillator by writing a 0 to the TD_OSCSEL bit.
3. Wait 5 slow clock cycles for internal resynchronization.

33.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SCKC_CR	31:24								TD_OSCSEL
		23:16								
		15:8								
		7:0								

33.5.1 Slow Clock Controller Configuration Register

Name: SCKC_CR
Offset: 0x0
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
								TD_OSCSEL
Access								R/W
Reset								0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 24 – TD_OSCSEL Timing Domain Slow Clock Selector

Value	Description
`0 (RC)`	Slow clock of the timing domain is driven by the embedded always-on 64 kHz (typical) RC oscillator.
`1 (XTAL)`	Slow clock of the timing domain is driven by the 32.768 kHz crystal oscillator.

34. Clock Generator

34.1 Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in the Register Summary. However, the Clock Generator registers are named CKGR_.

34.2 Embedded Characteristics

The Clock Generator is made up of:

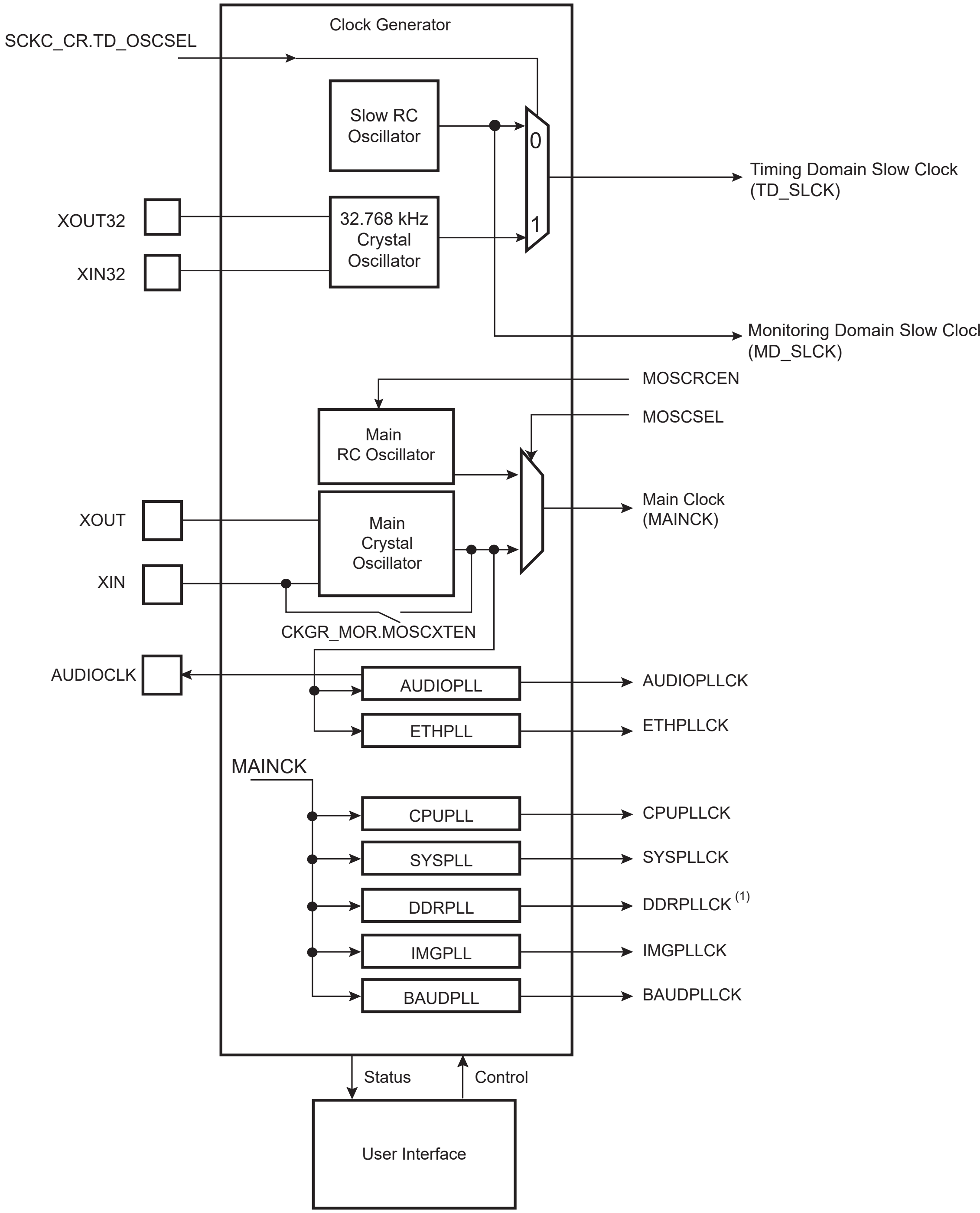
- Oscillators
  - A low-power 32.768 kHz oscillator supporting crystals and resonators (referred to as “32.768 KHz crystal oscillator” throughout the document)
  - An embedded always-on, slow RC oscillator generating a typical 32 kHz clock
  - A 12 to 50 MHz oscillator supporting crystals, resonators and Bypass mode (referred to as “main crystal oscillator” throughout the document)
  - A main RC oscillator generating a typical 12 MHz clock
- A set of 7 fractional-N PLLs with an input frequency range of 12 to 50 MHz and an internal frequency range of 600 to 1200 MHz, namely:
  - CPUPLL - PLL ID0
  - SYSPLL - PLL ID1
  - DDRPLL - PLL ID2
  - IMGPLL - PLL ID3
  - BAUDPLL - PLL ID4
  - AUDIOPLL - PLL ID5
  - ETHPLL - PLL ID6

The Clock Generator provides the following clocks:

- MD_SLCK—Monitoring domain slow clock. This clock, sourced from the always-on slow RC oscillator only, is the only permanent clock of the system and feeds safety-critical functions of the device (WDT, RSTC, SCKC, frequency monitors and detectors, PMC start-up time counters).
- TD_SLCK—Timing domain slow clock. This clock, sourced from the 32.768 kHz crystal oscillator or the always-on slow RC oscillator, is routed to the RTC and RTT peripherals.
- MAINCK—Output of the main clock oscillator selection. This clock is either the main crystal oscillator or the main RC oscillator.
- PLL Clocks—Outputs of embedded PLLs

34.3 Block Diagram

Figure 34-1. Clock Generator Block Diagram



Note (1) DDRPLLCK is hard-wired to the DDR subsystem (MCK2).

34.4 Slow Clock

The PMC does not control the slow clock generation. The control of the slow clock is performed by the Slow Clock Controller (SCKC) which embeds a slow clock generator that is supplied with the VDDBU power supply. As soon as VDDBU is supplied, both the 32.768 kHz crystal oscillator and the slow RC oscillator are powered, but only the slow RC oscillator is enabled.

MD_SLCK is always generated by the slow RC oscillator.

TD_SLCK is generated either by the 32.768 kHz crystal oscillator or by the slow RC oscillator.

The TD_SLCK source clock selection is made via the TD_OSCSEL bit in the Slow Clock Controller Configuration register (SCKC_CR).

34.4.1 Slow RC Oscillator (32 kHz typical)

The slow RC oscillator is a permanent clock that is the source clock of MD_SLCK and the default source clock of TD_SLCK.

Compared to the 32.768 kHz crystal oscillator, this oscillator offers a faster start-up time and is less exposed to the external environment, as it is fully integrated. However, its output frequency is subject to larger variations with supply voltage, temperature and manufacturing process. Therefore, the user must take these variations into account when this oscillator is used as a time base (start-up counter, frequency monitor, etc.). Refer to the section “Electrical Characteristics”.

34.4.2 32.768 kHz Crystal Oscillator

By default, the 32.768 kHz oscillator is enabled. To use this oscillator, the XIN32 and XOUT32 pins must be connected to a 32.768 kHz crystal or to a ceramic resonator. Refer to the section “Electrical Characteristics” for appropriate loading capacitors selection on XIN32 and XOUT32.

To select the 32.768 kHz crystal oscillator as the source of TD_SLCK, SCKC_CR.TD_OSCSEL must be set. This results in a sequence which first configures the PIO lines multiplexed with XIN32 and XOUT32 and driven by the crystal oscillator, and then enables the 32.768 kHz crystal oscillator. The switch of TD_SLCK source is glitch-free.

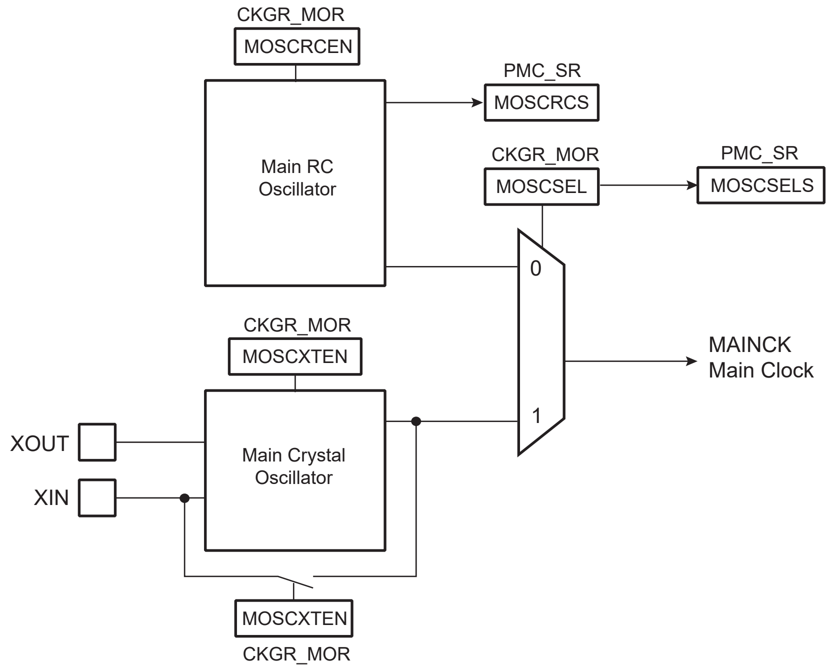
Reverting to the slow RC oscillator is only possible by shutting down the VDDBU power supply.

34.5 Main Clock

The main clock (MAINCK) has two sources:

- A main RC oscillator with a fast start-up time and that is selected by default to start the system
- A main crystal oscillator with Bypass mode

Figure 34-2. Main Clock (MAINCK) Block Diagram



34.5.1 Main RC Oscillator

After reset, the main RC oscillator is enabled. This oscillator is selected as the source of MAINCK. MAINCK is the default clock selected to start the system.

The main RC oscillator is calibrated in production. For output frequency specifications, refer to the section “Electrical Characteristics”.

The software can disable or enable the main RC oscillator with the MOSCRCEN bit in the Clock Generator Main Oscillator register (CKGR_MOR).

When disabling the main RC oscillator by clearing the CKGR_MOR.MOSCRCEN bit, the PMC_SR.MOSCRCS bit is automatically cleared, indicating that the oscillator is off.

Setting the MOSCRCS bit in the Power Management Controller Interrupt Enable Register (PMC_IER) triggers an interrupt to the processor.

34.5.2 Main Crystal Oscillator

After reset, the main crystal oscillator is disabled and is not selected as the source of MAINCK.

The software enables or disables this oscillator in order to reduce power consumption via CKGR_MOR.MOSCXTEN.

When disabling this oscillator by clearing CKGR_MOR.MOSCXTEN, the PMC_SR.MOSCXTS status bit is automatically cleared, indicating the oscillator is off. To activate the Main Crystal Oscillator Bypass mode, see Bypassing the Main Crystal Oscillator.

When enabling this oscillator, the user must initiate the start-up time counter. The start-up time depends on the characteristics of the external device connected to this oscillator.

When CKGR_MOR.MOSCXTEN and CKGR_MOR.MOSCXTST are written to enable this oscillator, XIN and XOUT are driven by the main crystal oscillator. PMC_SR.MOSCXTS is cleared and the counter starts counting down on MD_SLCK divided by 8 from the CKGR_MOR.MOSCXTST value. Since the CKGR_MOR.MOSCXTST value is coded with 8 bits, the start-up time can be programmed up to 2048 MD_SLCK periods, corresponding to about 62 ms when running at 32.768 kHz.

When the start-up time counter reaches ‘0’, PMC_SR.MOSCXTS is set, indicating that the oscillator is stabilized. Setting the MOSCXTS bit in the Interrupt Mask Register (PMC_IMR) can trigger an interrupt to the processor.

34.5.3 Main Clock Source Selection

The source of MAINCK can be selected from the following:

- the main RC oscillator
- the main crystal oscillator
- an external clock signal provided on the XIN input (Bypass mode of the main crystal oscillator)

The advantage of the main RC oscillator is its fast start-up time. By default, this oscillator is selected to start the system and it must be selected prior to entering ULP1 or ULP2 mode.

The advantage of the main crystal oscillator is its high level of accuracy.

The selection of the oscillator is made by configuring CKGR_MOR.MOSCSEL. The switchover of the MAINCK source is glitch-free, thus the switchover can be performed even if MCK0 is fed by MAINCK. PMC_SR.MOSCSELS indicates when the switch sequence is done.

Setting PMC_IMR.MOSCSELS triggers an interrupt to the processor.

34.5.4 Bypassing the Main Crystal Oscillator

Prior to bypassing the main crystal oscillator, the XOUT pin must be grounded and the external clock frequency provided on the XIN pin must be stable and within the values specified in the XIN Clock characteristics in the section “Electrical Characteristics”. Then the main crystal oscillator must be enabled by setting the CKGR_MOR.MOSCXTEN bit to 1.

34.5.5 Main Frequency Counter

The main frequency counter measures the main RC oscillator or the main crystal oscillator against the MD_SLCK and is managed by CKGR_MCFR.

During the measurement period, the main frequency counter increments at the speed of the clock defined by the bit CKGR_MCFR.CCSS.

A measurement is started in the following cases:

- When CKGR_MCFR.RCMEAS is written to ‘1’.
- When the main RC oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when the MOSCRCS bit is set)
- When the main crystal oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when the MOSCXTS bit is set)
- When MAINCK source selection is modified

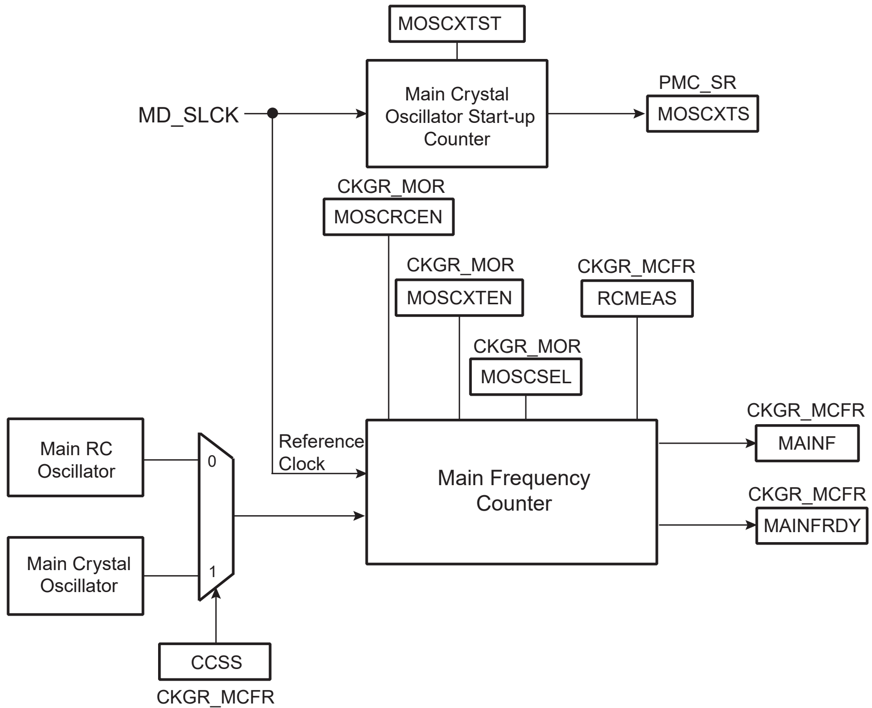
The measurement period ends at the 16th falling edge of MD_SLCK, the MAINFRDY bit in CKGR_MCFR is set and the counter stops counting. Its value can be read in the MAINF field of CKGR_MCFR and gives the number of clock cycles during 16 periods of MD_SLCK, so that the frequency of the main RC oscillator or main crystal oscillator can be determined.

When switching the source of MAINCK from the main RC oscillator to the main crystal oscillator, follow the programming sequence below to ensure that the oscillator is present and that its frequency is valid:

1. Enable the main crystal oscillator by setting CKGR_MOR.MOSCXTEN. Configure the CKGR_MOR. MOSCXTST field with the main crystal oscillator start-up time as defined in the section “Electrical Characteristics”.
2. Wait for PMC_SR.MOSCXTS flag to rise, indicating the end of a start-up period of the main crystal oscillator.
3. Select the main crystal oscillator as the source clock of the main frequency counter by setting CKGR_MCFR.CCSS.
4. Initiate a frequency measurement by setting CKGR_MCFR.RCMEAS.
5. Read CKGR_MCFR.MAINFRDY until its value equals 1.
6. Read CKGR_MCFR.MAINF and compute the value of the main crystal frequency.

If the MAINF value is valid, software can switch MAINCK to the main crystal oscillator. See Main Clock Source Selection.

Figure 34-3. Main Frequency Counter Block Diagram



34.6 PLL Controls

The PMC embeds 7 PLLs that are controlled by the PMC_PLL_CTRL0, PMC_PLL_CTRL1, PMC_PLL_SSR, PMC_PLL_ACR and PMC_PLL_UPDATE registers. Each PLL is accessed in read or write through its index as defined in the table below, corresponding to the register field PMC_PLL_UPDT.ID. At any time, PLL_CTRL0, PLL_CTRL1 and PLL_ACR reflect the controls for the PLL with index PMC_PLL_UPDT.ID. When the UPDATE bit is set in PMC_PLL_UPDT, the PLL of index PMC_PLL_UPDT.ID is updated with the content of registers PLL_CTRL0, PLL_CTRL1 and PLL_ACR.

Although these PLLs are identical, their implementations differ in terms of input clock signal, maximum output clock frequency or availability of a dedicated output line.

Each PLL is fed by either the MAINCK or the main crystal oscillator and has a constraint on the frequency it can generate on its clock output. Refer to the section “Electrical Characteristics”.

The table below describes all PLLs with their names and source clocks. For maximum frequency, refer to the section “Electrical Characteristics”.

Table 34-1. PLL IDs

Index	PLL Name	Clock Name	PLL Clock Source	Usage Example	IO PLL Clock
0	CPUPLL	CPUPLLCK	MAINCK	CPU clock source	–
1	SYSPLL	SYSPLLCK	MAINCK	MCK1 clock source (matrixes, etc.)	–
2	DDRPLL	DDRPLLCK	MAINCK	Clock source (DDR Phy)	–
3	IMGPLL	IMGPLLCK	MAINCK	MCK3 clock source (Image subsystem)	–
4	BAUDPLL	BAUDPLLCK	MAINCK	GCLK source for FLEXCOM, SDMMC, etc.	–

...........continued					
Index	**PLL Name**	**Clock Name**	**PLL Clock Source**	**Usage Example**	**IO PLL Clock**
5	AUDIOPLL	AUDIOPLLCK	MAIN XTAL OSC	AUDIOCLK output clock source	AUDIO_CLK
6	ETHPLL	ETHPLLCK	MAIN XTAL OSC	GMAC GCLK clock source	–

34.6.1 Divider and Phase Lock Loop Programming

Each PLL is controlled the same way. The internal clock frequency is configured by setting PMC_PLL_CTRL1.MUL and PMC_PLL_CTRL1.FRACR, then two division ratios are applied on the internal PLL clock:

- The first division ratio generates the PLL clock for the PMC.
- The second division ratio generates a clock for a dedicated output line (IOPLLCKx). This feature generates an output clock signal with improved jitter performance. It is not available for all PLLs. Refer to PLL Characteristics in the section “Electrical Characteristics” for more information.

The COREPLLCK operating frequency is defined as:

$$f_{\mathrm{COREPLLCK}} = f_{\mathrm{ref}}\left(\mathrm{MUL} + 1 + \frac{\mathrm{FRACR}}{2^{22}}\right)$$

The PLL clock frequency is defined by the following formula:

$$f_{\mathrm{PLL\ Clock}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVPMC} + 1)}$$

The IOPLLCK frequency is defined by the following formula:

$$f_{\mathrm{IOPLLCK}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVIO}+1)}$$

Each PLL sends a lock signal to the PMC to indicate its lock status. Once the lock signal has risen, the clock generated by the PLL is stable and can be sent to the PMC and/or its corresponding I/O if available.

This signal reports the lock status of the PLL by setting the corresponding PMC_PLL_CTRL0.ENLOCK to ‘1’.

If the lock status is disabled, a start-up time can be used instead in the PMC_PLL_UPDT register. The start-up time is expressed as a number of MD_SLCK cycles. Once the counter has reached the specified value, a flag rises. The start-up time field can only be written while all PLLs are disabled (i.e., their PLLEN fields are null).

If both a start-up time and the lock are enabled, the lock sent by the PLL is read once the start-up time has elapsed.

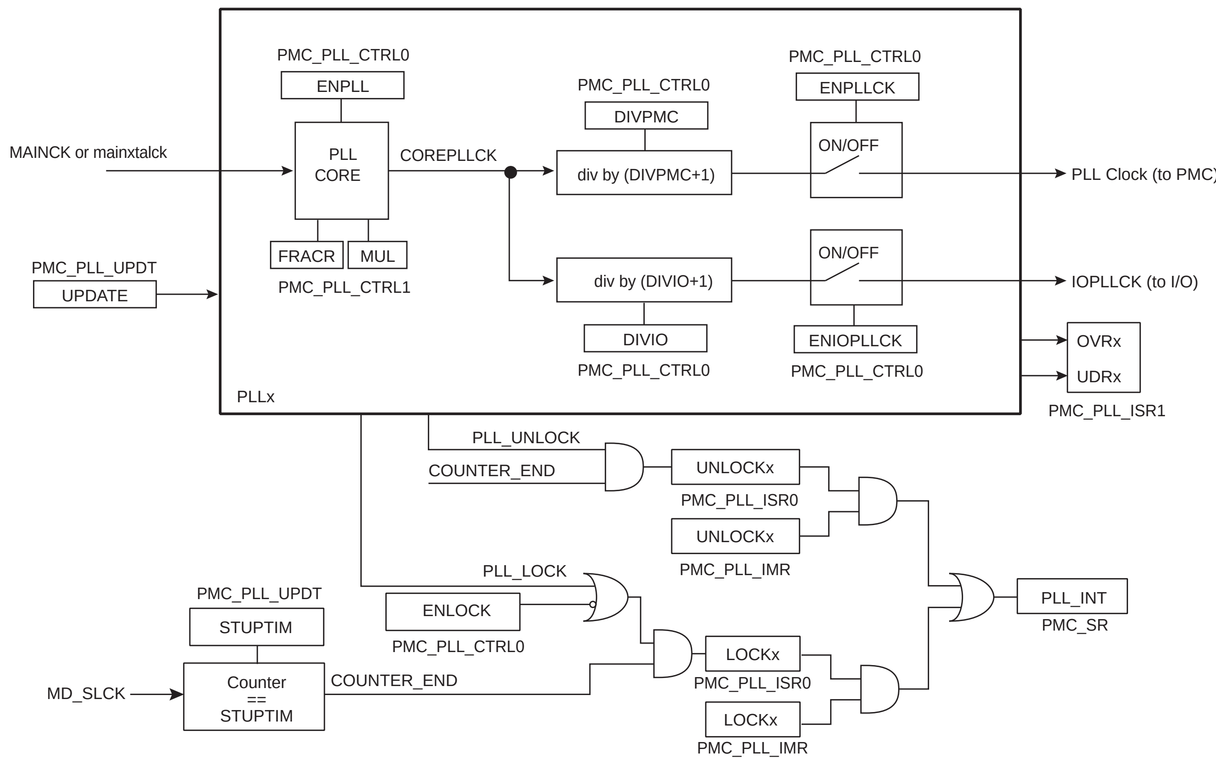
If neither the start-up time nor the lock are enabled, there is no way to know the lock status of the PLL.

The PLL also embeds an unlock status that informs when the PLL lock is lost. When enabled, this status is read once the start-up time (if defined) has elapsed.

The lock and unlock status can be used as interrupts.

See the following figure.

Figure 34-4. PLL Controls



Follow the steps below to enable a PLL:

1. Define the ID (ID=n) and start-up time by configuring the fields PMC_PLL_UPDT.ID and PMC_PLL_UPDT.STUPTIM. Set PMC_PLL_UPDT.UPDATE to ‘0’.
2. Configure PMC_PLL_ACR. See recommended values in the Electrical Characteristics section.
3. Define the MUL and FRACR to be applied to PLL(n) in PMC_PLL_CTRL1.
4. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1., otherwise the update is cancelled.
5. In PMC_PLL_CTRL0, write a ‘1’ to ENLOCK and to ENPLL and configure DIVPMC, DIVIO, ENPLLCK and ENIOPLLCK.
6. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1. otherwise the update is cancelled.
7. Wait for the lock bit to rise by polling the PMC_PLL_ISR0 or by enabling the corresponding interrupt in PMC_PLL_IER.
8. Disable the interrupt (if enabled).
9. Enable the unlock interrupt to quickly detect a failure on the generation of the PLL clock.

Once enabled (PMC_PLL_CTRL0.ENPLL=1), the PLL core generates its core clock (COREPLLCK).

Once the PLL has been enabled and has locked, the PLL configuration can be modified without switching off the cell.

The clock generated by the PLL is sent to the PMC if ENPLLCK is set to ‘1’ and the PMC_PLL_UPDT.UPDATE bit has then been written to ‘1’.

If the corresponding IOPLLCK is implemented, the clock generated by the PLL is sent to the IO if ENIOPLLCK was set to 1 and then PMC_PLL_UPDT.UPDATE was written to 1.

To disable a PLL, the following sequence must be applied:

1. If the PLL drives a section of the system that is active, modify the source clock of the system.
2. Define the ID (ID=n) of the PLL to be switched off in PMC_UPDT. The bit UPDATE in this register must be set at 0 in this step.
3. In PMC_PLL_CTRL0, set ENPLLCK and ENIOPLLCK to 0 and leave ENPLL at ‘1’.

4. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during step 2, otherwise the update is cancelled.
5. Write a ‘0’ to PMC_PLL_CTRL0.ENPLL.
6. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 2., otherwise the update is cancelled.

34.6.2 UTMI PLL

UTMI PLLs are controlled in USB 2.0 PHYs. Refer to the section “USB 2.0 PHY”.

34.6.3 PLL Unlock

Each PLL has an unlock flag. It is recommended to set the UNLOCK interrupt by setting the corresponding PMC_PLL_IER.UNLOCK bit to quickly detect a failure on PLL clock generation.

The rise of a PLL unlock signal implies a failure in the normal operation of the PLL (input clock loss, for example). In this case, the PLL keeps operating but stops trying to lock the input clock. A manual clock switching to a stable clock should be performed to ensure CPU_CLK integrity

34.6.4 Spread Spectrum

Spread spectrum is obtained by slightly modifying the PLL target frequency. Two parameters are used to configure the spread spectrum:

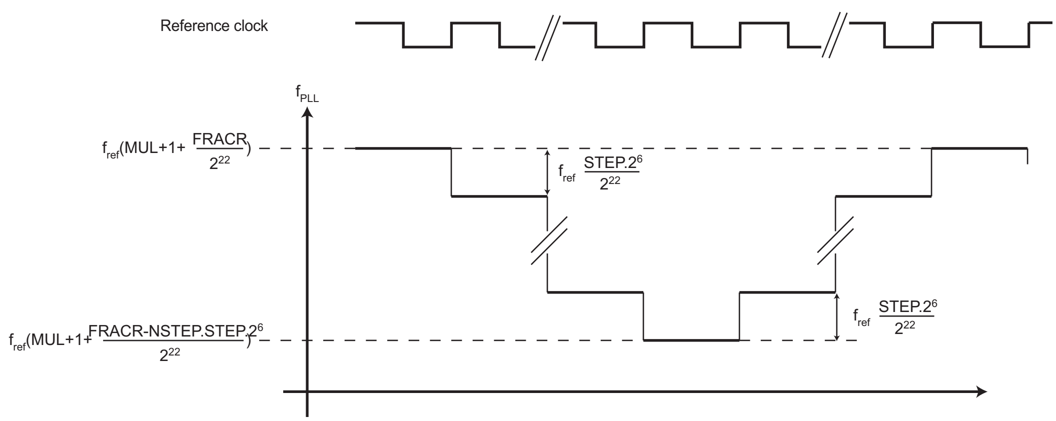
- STEP—the frequency step
- NSTEP—the number of times the STEP will be applied

The spread spectrum can be applied only if the PLL is already enabled and locked. Once the spread spectrum has been enabled, it is no longer possible to modify the target frequency of the PLL. Prior to change the PLL frequency, the spread spectrum must be disabled and a period of 2 x NSTEP cycles of the PLL source clock must elapse.

When enabled, the spread spectrum logic modifies the fractional part of the PLL. The fractional factor applied to the PLL is in the following range: FRACR - (64 x STEP x NSTEP) up to FRACR.

Starting from the base frequency of the PLL configured in PMC_PLL_CTRL1 (MUL, FRACR), the spread spectrum mechanism decreases the PLL frequency, and when the minimum is reached, the PLL frequency is increased up to the value configured through the PMC_PLL_CTRL1 register (the PLL frequency never overpasses that value).

Figure 34-5. Spread Spectrum Mechanism



35. Power Management Controller (PMC)

35.1 Description

The Power Management Controller (PMC) optimizes power consumption by controlling all system and user peripheral clocks. The PMC enables/disables the clock inputs to many of the peripherals and to the processor.

The Slow Clock Controller (SCKC) selects the source of TD_SLCK (drives the real-time part (RTT/RTC)). The source of MD_SLCK (drives the rest of the system controller: wake-up logic, watchdog, PMC, etc.) is always the slow RC oscillator.

For the out of ROM configuration, refer to the section “Boot Strategies”.

35.2 Embedded Characteristics

The Power Management Controller provides the following clocks:

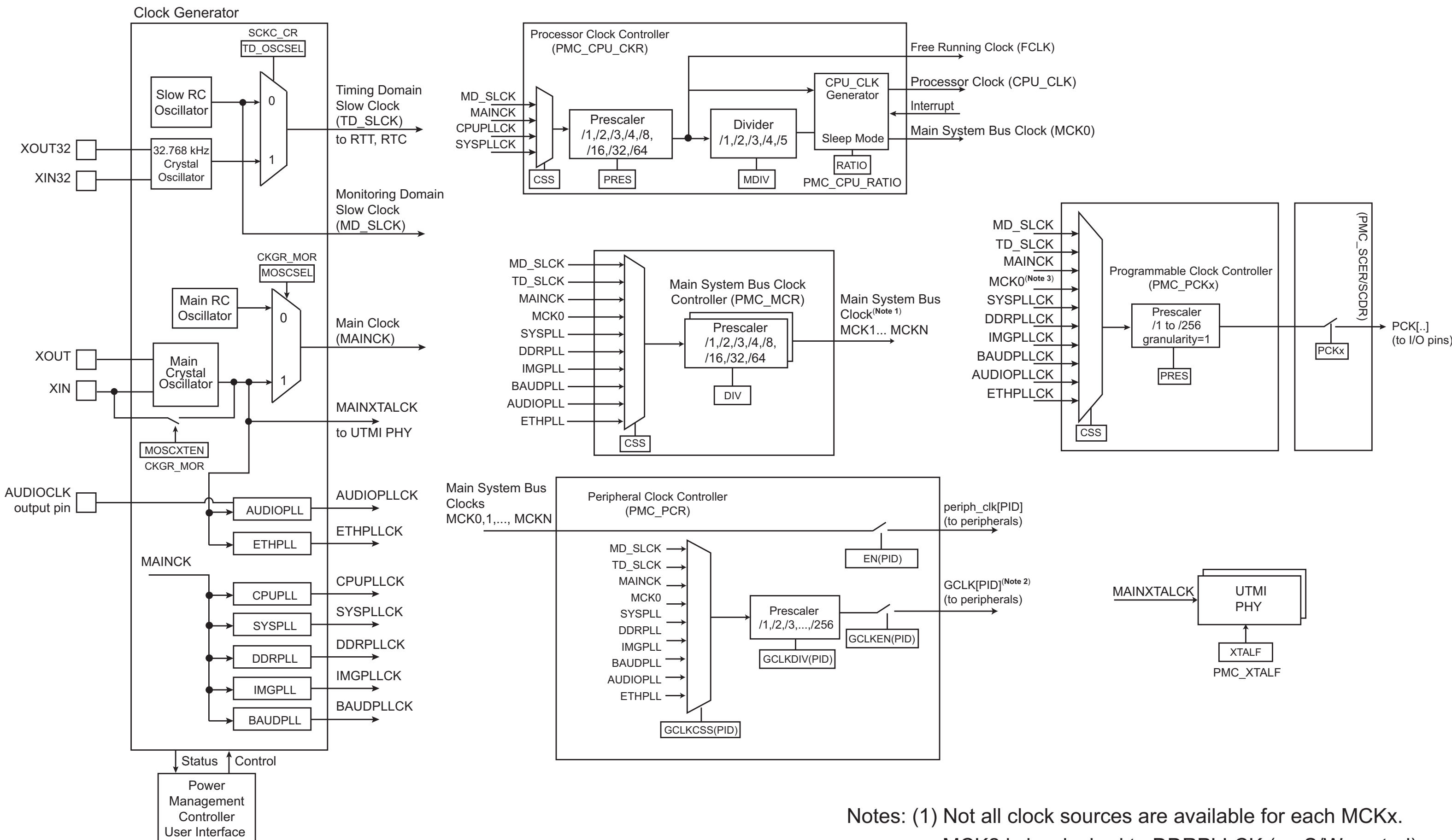
- Main System Bus Clocks (MCKx)—clock signals that are the root clock of a group of peripherals. They are programmable from a few hundred Hz to the maximum operating frequency of the peripheral group. MCK0 is the main system bus clock associated to all peripherals that are synchronous with the processor.
- Processor Clock (CPU_CLK)—can be tuned through a frequency scaler module and automatically switched off when entering the processor in Sleep mode.
- Free-running Processor Clock (FCLK)—the source clock of CPU_CLK. Is not affected when Sleep mode or the frequency scaler is activated.
- Peripheral Clocks with independent on/off control, provided to the peripherals. Each peripheral clock is inherited from one of the MCKx clocks.
- Programmable Clock Outputs (PCKx), selected from the clock generator outputs to drive the device PCKx pins.
- Generic Clock (GCLK) with controllable division and on/off control, independent of MCKx and CPU_CLK. Provided to selected peripherals. Refer to the table “Peripheral Identifiers” for more details on GCLK availability per peripheral.

The Power Management Controller also provides the following features on clocks:

- A main crystal oscillator failure detector
- A 32.768 kHz crystal oscillator frequency monitor
- A frequency counter on main crystal oscillator or main RC oscillator
- An MCK0 failure detector

35.3 Block Diagram

Figure 35-1. General Clock Distribution Block Diagram



Notes: (1) Not all clock sources are available for each MCKx.
MCK2 is hard-wired to DDRPLLCK (no S/W control).
(2) Not all clock sources are available for each GCLK[PID].
(3) As MCK0=CPU_CLK/PMC_CPU_CKR.MDIV,
MCK0 is not independent from CPU_CLK. Take this into account in case of CPU frequency scaling.

35.4 Processor Clock Controller

The PMC features a Processor Clock (CPU_CLK) controller that implements the processor Sleep mode. CPU_CLK can be disabled by executing the WFI (WaitForInterrupt) or the WFE (WaitForEvent) processor instruction.

CPU_CLK is enabled after a reset and is automatically re-enabled by any enabled interrupt. The processor Sleep mode is entered by disabling CPU_CLK, which is automatically re-enabled by any enabled interrupt, or by the reset of the product.

When processor Sleep mode is entered, the current instruction is finished before the CPU_CLK is stopped, but this does not prevent data transfers from other hosts of the system bus.

The clock selection is done in PMC_CPU_CKR.CSS.

The prescaler is configured in PMC_CPU_CKR.PRES.

The Processor Clock Controller also generates a main system bus clock, MCK0, which is a subdivision of the FCLK.

Only one of CSS, PRES and MDIV fields can be modified at a time. When one of these parameters is modified, no other modification can be performed on these fields as long as the MCKRDY status flag is low.

Any modification in CSS, PRES or MDIV fields must never lead to generate an MCK frequency that is greater than the maximum allowed system frequency. When changing the source clock of the system to a faster clock, the fields must be modified using the following order: MDIV, PRES and then CSS. When changing the source clock of the system to a slower clock, the fields must be modified using the following order: CSS, PRES and then MDIV.

If the destination clock does not exist, the switching is not performed. The CPU_CLK keeps running with the previous clock and the system must be reset to run correctly again.

The user must implement a timeout when monitoring the MCKRDY flag after a clock source switching in order to perform a reset in case switching fails.

The Processor Clock Controller features a dynamic frequency scaling mechanism that allows the CPU_CLK to be a non-integer division of the free-running clock.

35.4.1 Processor Clock Ratio

The Processor Clock frequency can be controlled more precisely with the PMC_CPU_RATIO.RATIO.

$$f_{\mathrm{CPU}} = \frac{(\mathrm{RATIO} + 1)}{16} \times f_{\mathrm{FCLK}}$$

By default, RATIO = 15, and the Processor Clock frequency equals the Free-running Clock frequency. Modifying the RATIO field modifies the frequency of the Processor Clock with a frequency step of $f_{FCLK}/16$.

35.5 Free-running Processor Clock

The free-running Processor clock (FCLK) used for sampling interrupts and clocking debug blocks ensures that interrupts can be sampled, and sleep events can be traced, while the processor is sleeping.

35.6 Main System Bus Clock Controller

The Main System Bus Clock Controller provides the main system bus clocks (MCKx) of the different peripheral groups, with the exception of MCK0 that is generated by the Processor Clock Controller and MCK2 that is hard-wired to DDRPLLCK. MCKx are the source clocks of the peripheral clocks. Each peripheral clock is generated by one MCKx. To know which peripherals are clocked by a specific MCKx, refer to the Peripheral Identifiers table in the section "System Interconnect and Security (SIS)".

When configuring an MCKx, it is mandatory to write PMC_MCR.CMD to ‘1’ and PMC_MCR.ID with the index of the corresponding MCKx.

To read the current configuration of an MCKx, PMC_MCR must be first accessed with PMC_MCR.CMD written to ‘0’ and PMC_MCR.PID field written with the index of the corresponding MCKx. This write does not modify the configuration of the peripheral. The PMC_MCR can then be read to know the configuration status of the corresponding PID.

The clock selection is done in PMC_MCR.CSS.

The prescaler is configured in PMC_MCR.PRES.

Each time PMC_MCR.CSS is configured to define a new MCKx, PMC_SR.MCKXRDY is cleared. It reads ‘0’ until all MCKx are established. Then, MCKXRDY is set and can trigger an interrupt to the processor.

Table 35-1. Available Input for Each MCKx

Clock Sources	Main System Bus Clocks				
	MCK0	MCK1	MCK2	MCK3	MCK4
MD_SLCK	X	X	–	X	X
MAINCK	X	X	–	X	X
MCK0	–	X	–	X	X
SYSPLLCK	X	X	–	X	X

...........continued

Clock Sources	Main System Bus Clocks				
	MCK0	MCK1	MCK2	MCK3	MCK4
ETHPLLCK	–	–	–	–	–
IMGPLLCK	–	–	–	X	–
DDRPLLCK	–	–	X	X	–
CPUPLLCK	X	–	–	–	–

35.7 Peripheral and Generic Clock Controller

The PMC controls the clocks of the embedded peripherals by means of the Peripheral Control register (PMC_PCR). With this register, the user can enable and disable the different clocks used by the peripherals:

- Peripheral clocks (periph_clk[PID]), routed to every peripheral and derived from the corresponding MCKx. It is mandatory to enable this clock before using a peripheral.
- Generic clocks (GCLK[PID]), routed to selected peripherals only (refer to the “Peripheral Identifiers” table). These clocks are independent of the core and bus clocks (CPU_CLK, MCKx and periph_clk[PID]). They are generated by selection and division of available sources. The list of available source clocks depends on the peripheral. Refer to the description of each peripheral to know available sources and limitations to be applied to GCLK[PID] compared to periph_clk[PID].

To configure a peripheral’s clocks, PMC_PCR.CMD must be written to ‘1’ and PMC_PCR.PID must be written with the index of the corresponding peripheral. All other configuration fields must be correctly set.

To read the current clock configuration of a peripheral, PMC_PCR must be first accessed with PMC_PCR.CMD bit written to ‘0’ and PMC_PCR.PID field written with the index of the corresponding peripheral. This write does not modify the configuration of the peripheral. The PMC_PCR can then be read to know the configuration status of the corresponding PID.

The status of the peripheral clock activity can be read in the Peripheral Clock Status registers (PMC_CSRx).

The status of the peripheral generic clock activity can be read in the Generic Clock Status registers (PMC_GCSRx).

When a peripheral or a generic clock is disabled, it is immediately stopped. These clocks are disabled after a reset. The source and the division ratio of generic clocks must not be modified while the peripheral is enabled. The generic clock configuration must be set before the peripheral is enabled.

To stop a peripheral clock, it is recommended that the system software wait until the peripheral has executed its last programmed operation before disabling the clock. This is to avoid data corruption or erroneous behavior of the system.

35.8 Programmable Clock Output Controller

The PMC controls six signals to be output on the external pins PCKx. Each signal can be independently programmed via the Programmable Clock registers (PMC_PCKx).

PCKx can be independently selected between MD_SLCK, TD_SLCK, MAINCK, MCK0 and any PLLCK except PLLCK[0] by configuring PMC_PCKx.CSS. Each output signal can also be divided by 1 to 256 by configuring PMC_PCKx.PRES.

Each output signal can be enabled and disabled by writing a ‘1’ to the corresponding bits PMC_SCER.PCKx and PMC_SCDR.PCKx, respectively. The status of the active programmable output clocks is given in PMC_SCSR.PCKx.

The status flag PMC_SR.PCKRDYx indicates that the clock configured through the PMC_PCKx register is correctly established.

As the Programmable Clock Controller does not manage with glitch prevention when switching clocks, it is strongly recommended to disable PCKx before any configuration change and to re-enable it once the change is performed.

35.9 Ultra-Low Power Modes and Fast Start-Up

35.9.1 ULP1 and ULP2 Modes

Refer to the section “Electrical Characteristics”.

35.9.2 Fast Start-Up

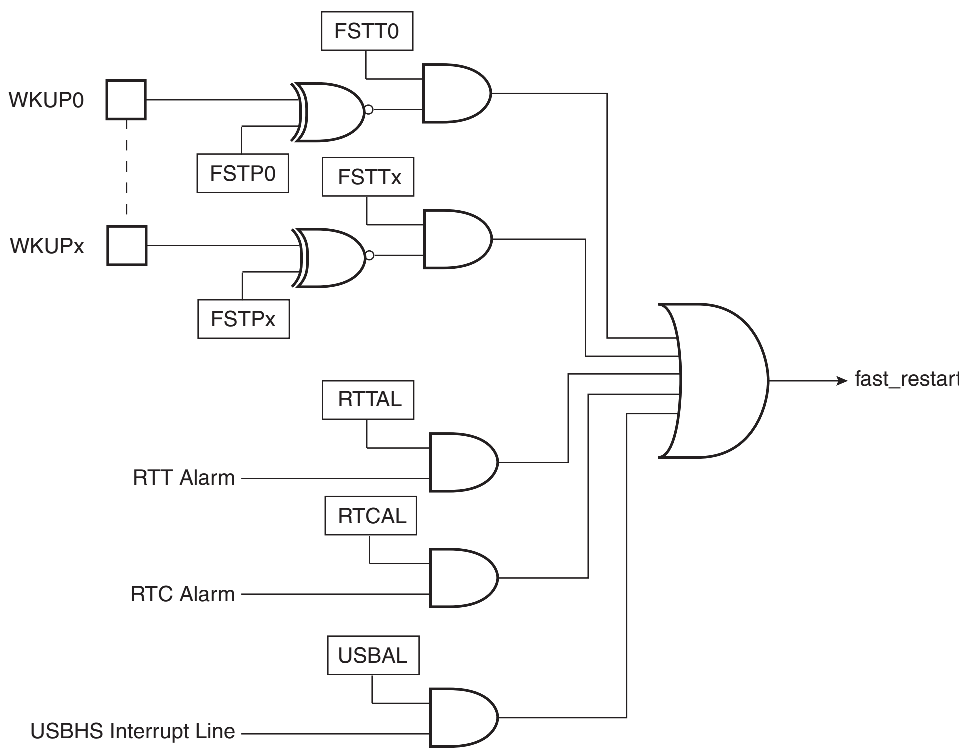
At exit from ULP1 or ULP2 mode, the device allows the processor to restart in several microseconds only if the C-code function that manages the ULP1 or ULP2 mode entry and exit is linked to and executed from on-chip SRAM.

A fast start-up occurs upon the detection of a programmed level on one of the wake-up inputs (WKUP) or upon an active alarm from the RTC, RTT and USB Controller. The polarity of each of the wake-up inputs is programmable in the PMC Wake-Up Control Register (PMC_WCR).

⚠WARNING The duration of the WKUPx pins active level must be greater than four MAINCK cycles.

The fast start-up circuitry, as shown in the following figure, is fully asynchronous and provides a fast start-up signal to the PMC. As soon as the fast start-up signal is asserted, the main RC oscillator restarts automatically.

Figure 35-2. Fast Start-Up Circuitry



Each wake-up input pin can be configured to generate a fast start-up event by setting the corresponding bits in PMC_WCR.

To configure a wake-up pin, a write access must be performed in PMC_WCR (CMD=’1’). Field PID must be written with the ID of the wake-up pin, FSTP set to the polarity of the wake-up pin and EN set to enable/disable the wake-up pin.

To read the configuration status of a wake-up pin, PMC_WCR.PID must be written with the ID of the wake-up pin and the CMD bit set to ‘0’. Then the next read access to PMC_WCR sends the configuration status of the wake-up pin specified in PID.

Each alarm can be enabled to generate a fast start-up event by setting the corresponding bit in PMC_FSMR.

The user interface does not provide any status for fast start-up. The status can be read in the PIO Controller and the status registers of the RTC, RTT and USB Controller.

35.10 Asynchronous Partial Wake-Up

35.10.1 Description

The asynchronous partial wake-up wakes up a peripheral in a fully asynchronous way when activity is detected on the communication line. The asynchronous partial wake-up function automatically manages the peripheral clock. It reduces overall power consumption of the system by clocking peripherals only when needed.

Asynchronous partial wake-up can be enabled in ULP1, ULP2 or Active mode.

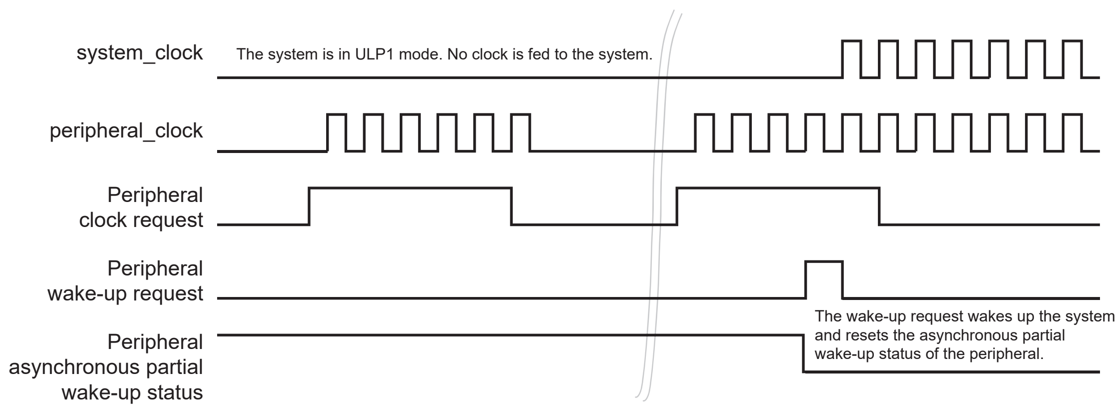
Only the following peripherals can be configured with asynchronous partial wake-up: FLEXCOMx and ADCC.

The peripheral selected for asynchronous partial wake-up must first be configured so that its clock is enabled. To do so, configure PMC_PCR.PID and write a ‘1 to PMC_PCR.EN.

35.10.2 Asynchronous Partial Wake-Up when System is in ULP1 Mode

When an asynchronous clock request from a peripheral occurs, the PMC partially wakes up the system to feed the clock only to this peripheral. The rest of the system is not fed with the clock, thus optimizing power consumption. Finally, depending on user-configurable conditions, the peripheral either wakes up the whole system if these conditions are met or stops the peripheral clock until the next clock request. If a wake-up request occurs, asynchronous partial wake-up is automatically disabled until the user instructs the PMC to enable it. This is done by writing a ‘1’ to SLPWKSR in the Asynchronous Partial Wake-Up Control register (PMC_SLPWKCR).

Figure 35-3. System Asynchronous Partial Wake-Up Waveforms



35.10.2.1 Configuration Procedure

Before configuring asynchronous partial wake-up for a peripheral, check that the PIDx bit in PMC_CSR is set. This ensures that the peripheral clock is enabled.

The steps to enable asynchronous partial wake-up for a peripheral are the following:

1. Check that PMC_SLPWKCR.ASR is set to ‘0’ for the corresponding peripheral. This ensures that the peripheral has no activity in progress.
2. Enable asynchronous partial wake-up for the peripheral by writing a ‘1’ to PMC_SLPWKCR.SLPWKSR.
3. Check that PMC_SLPWKCR.ASR is set to '0'. This ensures that no activity has started during the enable phase.
   If PMC_SLPWKCR.ASR=1, proceed to the next step.

   If PMC_SLPWKCR.ASR=0, Asynchronous Partial Wake-up mode is active for the peripheral. Before entering ULP1 or ULP2 mode, check that the AIP bit in the Asynchronous Partial Wake-Up Activity In Progress register (PMC_SLPWK_AIPR) is cleared. This ensures that the peripheral has no activity in progress due to a reactivation.
4. In PMC_SLPWKCR, asynchronous partial wake-up must be immediately disabled by writing a ‘0’ to SLPWKSR. Wait for the end of peripheral activity before reinitializing the procedure.

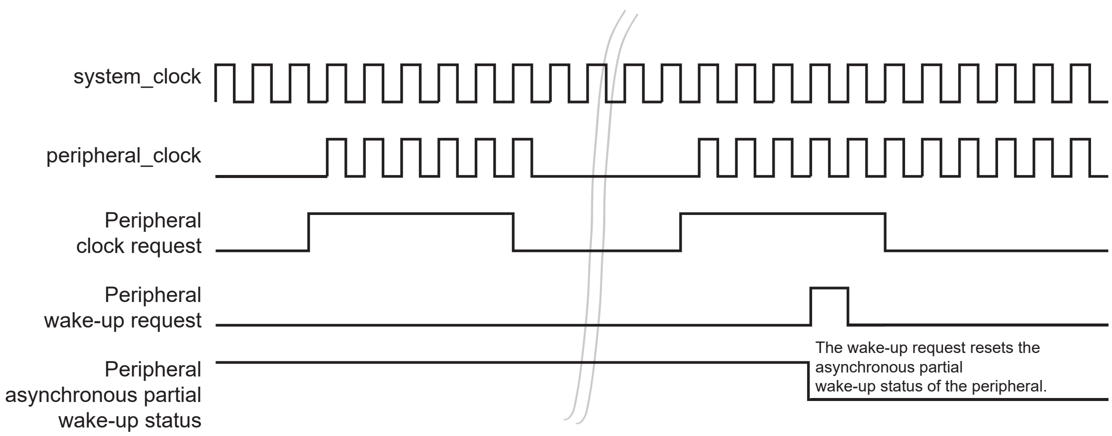
35.10.3 Asynchronous Partial Wake-Up of a Peripheral in Active Mode

When the system is in Active mode, peripherals enabled for asynchronous partial wake-up have their respective clocks stopped until the peripherals request a clock. When a peripheral requests the clock, the PMC provides the clock without processor intervention.

The triggering of the peripheral clock request depends on conditions which can be configured for each peripheral. If these conditions are met, the peripheral asserts a request to the PMC. The PMC disables the Asynchronous Partial Wake-up mode of the peripheral and provides the clock to the peripheral until the user instructs the PMC to re-enable partial wake-up on the peripheral. This is done by setting PMC_SLPWKCR.SLPWKSR.

If the conditions are not met, the peripheral clears the clock request and the PMC stops the peripheral clock until the clock request is reasserted by the peripheral.

Figure 35-4. Asynchronous Partial Wake-up in Active Mode



35.10.3.1 Configuration Procedure

Before configuring the asynchronous partial wake-up function of a peripheral, check that the PIDx bit in PMC_CSR is set. This ensures that the peripheral clock is enabled.

The steps to enable the asynchronous partial wake-up function of a peripheral are the following:

1. Check that PMC_SLPWKCR.ASR is set to ‘0’. This ensures that the peripheral has no activity in progress.
2. Enable the asynchronous partial wake-up function of the peripheral by writing a ‘1’ to PMC_SLPWK_ER.SLPWKSR.
3. Check that PMC_SLPWKCR.ASR is set to '0'. This ensures that no activity has started during the enable phase.
   If PMC_SLPWKCR.ASR=1, proceed to the next step.

   If PMC_SLPWKCR.ASR=0, Asynchronous Partial Wake-up mode is active for the peripheral. Before entering ULP1 or ULP2 mode, check that the AIP bit in the PMC Asynchronous Partial Wake-Up Activity In Progress register (PMC_SLPWK_AIPR) is cleared. This ensures that the peripheral has no activity in progress due to a reactivation.
4. In PMC_SLPWKCR, asynchronous partial wake-up must be immediately disabled by writing a ‘0’ to SLPWKSR. Wait for the end of peripheral activity before reinitializing the procedure.

Note: If an activity has started during the enable phase, the asynchronous partial wake-up function must be immediately disabled by writing a ‘0’ to PMC_SLPWKCR.SLPWKSR. Wait for the end of peripheral activity before reinitializing the procedure.When asynchronous partial wake-up for a peripheral is enabled and the core is running (system not in ULP1 or ULP2 mode), the peripheral must not be accessed before a wake-up of the peripheral is performed.

35.11 Main Crystal Oscillator Failure Detection

The main crystal oscillator failure detector monitors the main crystal oscillator against the slow RC oscillator and provides an automatic switchover of the MAINCK source to the main RC oscillator in case of failure detection.

The failure detector can be enabled or disabled by configuring CKGR_MOR.CFDEN. It cannot be enabled if the main crystal oscillator is disabled. It must be disabled before disabling the main crystal oscillator.

It is also disabled in either of the following cases:

- after a VDDCORE reset
- when the main crystal oscillator is disabled (MOSCXTEN = 0)

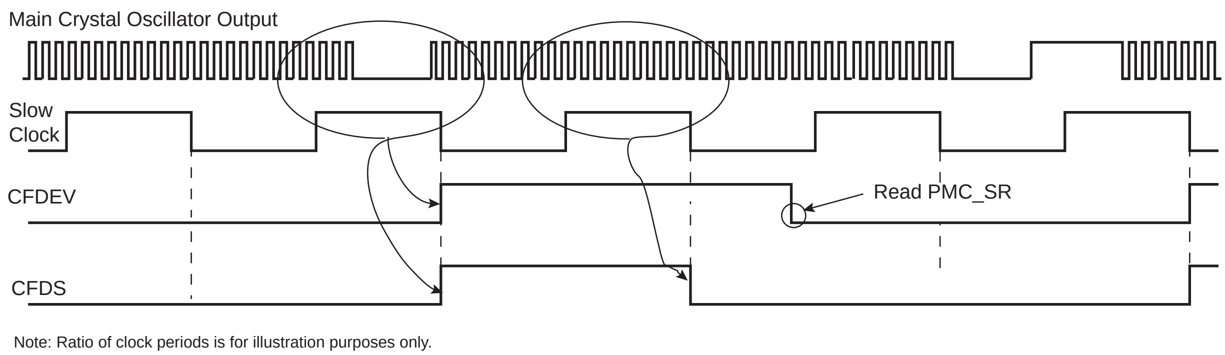
A failure is detected by means of a counter incrementing on the main crystal oscillator output and detection logic is triggered by the slow RC oscillator which is automatically enabled when CFDEN = 1.

The counter is cleared when the slow RC oscillator clock signal is low and enabled when the signal is high. Thus, the failure detection time is one slow RC oscillator period. If, during the high level period of the slow RC oscillator clock signal, less than eight main crystal oscillator clock periods have been counted, then a failure is reported. Note that when enabling the failure detector, up to two cycles of the slow RC oscillator are needed to detect a failure of the main crystal oscillator.

If a main crystal oscillator failure is detected, PMC_SR.CFDEV and PMC_SR.FOS both indicate a failure event. PMC_SR.CFDEV is cleared on read of PMC_SR, and PMC_SR.FOS is cleared by writing a ‘1’ to the FOCLR bit in the PMC Fault Output Clear register (PMC_FOCR).

Only PMC_SR.CFDEV can generate an interrupt if the corresponding interrupt source is enabled in PMC_IER. The current status of the clock failure detection can be read at any time from PMC_SR.CFDS.

Figure 35-5. Clock Failure Detection Example



If the CKGR_MOR.AUTOMAINSW bit is set to'1', the source of MAINCK automatically switches to the MAIN RC oscillator. If the main RC oscillator was previously powered off, it is first powered on before switching. If the CKGR_MOR.AUTOCPUSW bit is set to'1', the source of MCK0 automatically switches to MAINCK.

If the main crystal oscillator is selected as the source of MAINCK, the PMC can be configured to automatically select the main RC oscillator as the source of MAINCK in case of a main crystal oscillator failure detection by setting the CKGR_MOR.AUTOMAINSW to '1'. Additionally, if the source of CPU_CLK is a PLL driven by the main crystal oscillator, the PMC can be configured to automatically select the MAINCK as the source of CPU_CLK in case of a main crystal oscillator failure detection by setting the CKGR_MOR.AUTOCPUSW to '1'. CKGR_MOR.AUTOMAINSW must be set to '1' prior to setting CKGR_MOR.AUTOCPUSW to '1'.

Two slow RC oscillator clock cycles are necessary to detect and switch from the main crystal oscillator to the main RC oscillator if the source of MCK0 is MAINCK, or three slow RC oscillator clock cycles if the source of MCK0 is a PLL.

A clock failure detection activates a fault output that is connected to the Pulse Width Modulator (PWM) controller. With this connection, the PWM controller is able to force its outputs and to protect the driven device if a clock failure is detected.

35.12 32.768 kHz Crystal Oscillator Frequency Monitor

The frequency of the 32.768 kHz crystal oscillator can be monitored by configuring CKGR_MOR.XT32KFME. Prior to enabling the monitoring, the 32.768 kHz crystal oscillator must be started and its start-up time be elapsed. Refer to the section “Slow Clock Controller (SCKC)” for details on the slow clock generator.

An error flag (PMC_SR.XT32KERR) is asserted when the 32.768 kHz crystal oscillator frequency is out of its nominal frequency value (i.e., 32.768 kHz). The error flag can be cleared only if the monitoring is disabled.

The frequency drift is computed with the main RC oscillator.

The monitored clock frequency is declared invalid if at least four consecutive 32.768 kHz crystal oscillator clock period measurement results are over the nominal period.

The error flag can be defined as an interrupt source of the PMC by setting PMC_IER.XT32KERR. This flag is also routed to the Reset Controller (RSTC) and may generate a reset of the device.

35.13 MCK0 Frequency Monitor

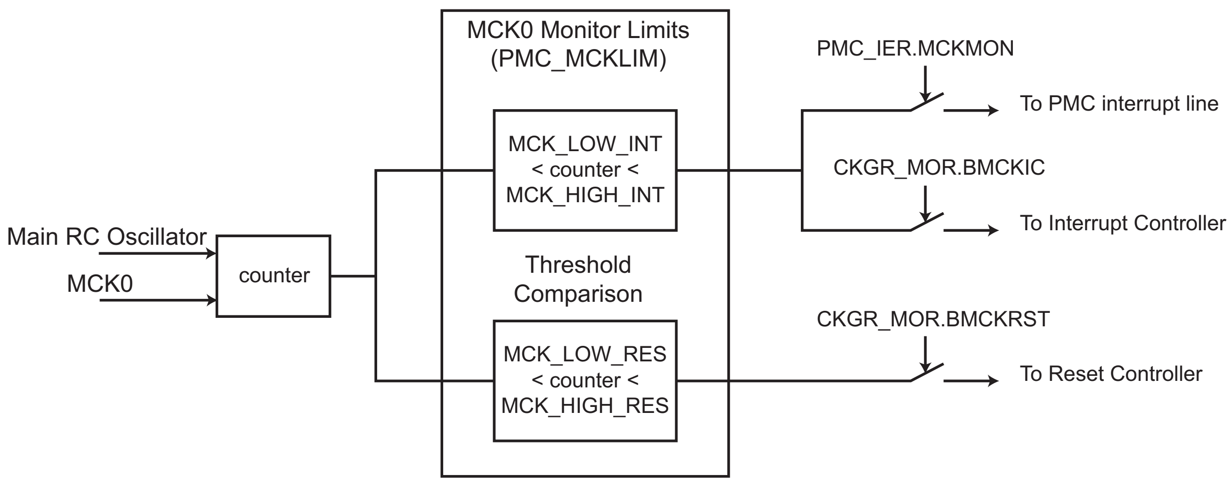
The frequency of MCK0 can be monitored with the main RC oscillator. This monitoring can only be performed if the MCK0 frequency is at least three times faster than the embedded main RC oscillator. This function is enabled by writing a ‘1’ to CKGR_MOR.BMCKIC, CKGR_MOR.BMCKRST or PMC_IER.MCKMON.

An error on the MCK0 frequency can lead to a PMC interrupt or a reset of the system (refer to the section “Reset Controller (RSTC)”).

When the corresponding PMC interrupt is enabled, the status of the MCK0 monitoring can be read on PMC_SR.MCKMON. This status is cleared on read.

Once enabled, the monitor continuously counts the number of MCK0 cycles within 15 cycles of the embedded main RC oscillator. The result is then compared to threshold values defined in the PMC_MCKLIM register. Two levels of thresholds can be defined: a first couple of values defines thresholds to generate an interrupt and a second couple of values defines thresholds to generate a reset of the system.

Figure 35-6. MCK0 Frequency Monitor



35.14 Recommended Programming Sequence

Follow the steps below to program the PMC:

1. If the main crystal oscillator is not required, the PLL can be directly configured (step 5) else this oscillator must be started (step 2).
2. Verify the existence and frequency value of the main crystal oscillator following the sequence defined in Main Frequency Counter
3. If the main crystal oscillator is enabled and valid, the source of MAINCK can be switched to the main crystal oscillator by writing CKGR_MOR.MOSCSEL to 1 else the PLL can be directly configured.
4. Wait for the end of the MAINCK source switching by either polling the MOSCSELS or setting the corresponding interrupt
5. Configure the PLLs by following the setup defined in Divider and Phase Lock Loop Programming (if not required, proceed to step 6):

6. Configure the MCK0 division ratio by setting PMC_CPU_CKR.MDIV. Available values are 0, 1, 2, 3, 4. MCK0 output is the CPU_CLK frequency divided by 1, 2, 3, 4 or 5, depending on the value programmed in MDIV. By default, MDIV is cleared, which indicates that the CPU_CLK is equal to MCK0.
7. Wait for the end of the MCK0 ratio switching by either polling the MCKRDY or setting the corresponding interrupt.
8. Select the division ratio of CPU_CLK by setting PMC_CPU_CKR.PRES.
   PRES is used to define the CPU_CLK and MCK0 prescaler. The user can choose between different values (1, 2, 3, 4, 8, 16, 32, 64). Prescaler output is the selected clock source frequency divided by the PRES value.
9. Wait for the end of the CPU_CLK ratio switching by either polling the MCKRDY or setting the corresponding interrupt.
10. Select the source clock of CPU_CLK by setting PMC_CPU_CKR.CSS.
    CSS is used to select the clock source of MCK0 and CPU_CLK. By default, the selected clock source is MAINCK.
11. Wait for the end of the CPU_CLK source switching by either polling the MCKRDY or setting the corresponding interrupt.
    PMC_CPU_CKR must not be programmed in a single write operation.

    Reconfiguring MDIV, PRES and CSS fields must always be done by following the right order of operation described above (steps 6 to 11).
12. Configure the prescaler of MCKx in PMC_MCR.PRES.
13. Configure the source of MCKx in PMC_MCR.CSS.
14. Configure the programmable clocks (PCKx):
    PCKx are controlled via registers PMC_SCER, PMC_SCDR and PMC_SCSR.

    PCKx can be enabled and/or disabled via PMC_SCER and PMC_SCDR. Three PCKx can be used. PMC_SCSR indicates which PCKx is enabled. By default all PCKx are disabled.

    PMC_PCKx registers are used to configure PCKx as described in Programmable Clock Output Controller.
15. Enable the peripheral and generic clocks.
    Once all of the previous steps have been completed, the peripheral and generic clocks can be configured via register PMC_PCR as described in Peripheral and Generic Clock Controller.

Note: As MCK0=CPU_CLK/PMC_CPU_CKR.MDIV, MCK0 is not independent from CPU_CLK. Take this into account in case of CPU frequency scaling.

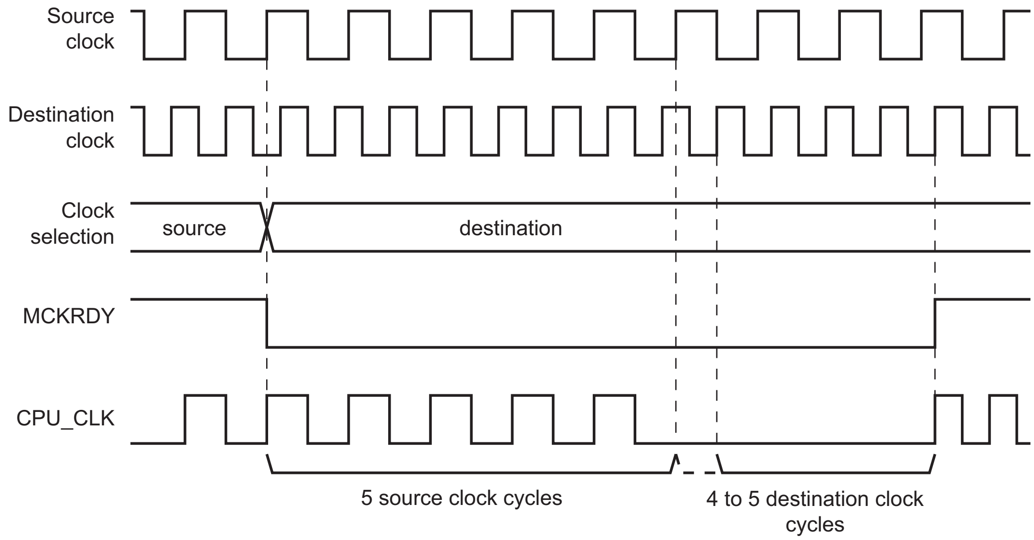
35.15 Clock Switching Details

35.15.1 CPU Clock Switching Timings

The glitch-free clock switcher implemented to control the sources of CPU_CLK and MCK0 performs clock switching in 5 clock cycles of the currently used clock plus 5 cycles of the target clock.

The clock switching is effective once MCKRDY rises. See the following figure.

Figure 35-7. Switch CPU Clock (CPU_CLK) from Source Clock to Destination Clock

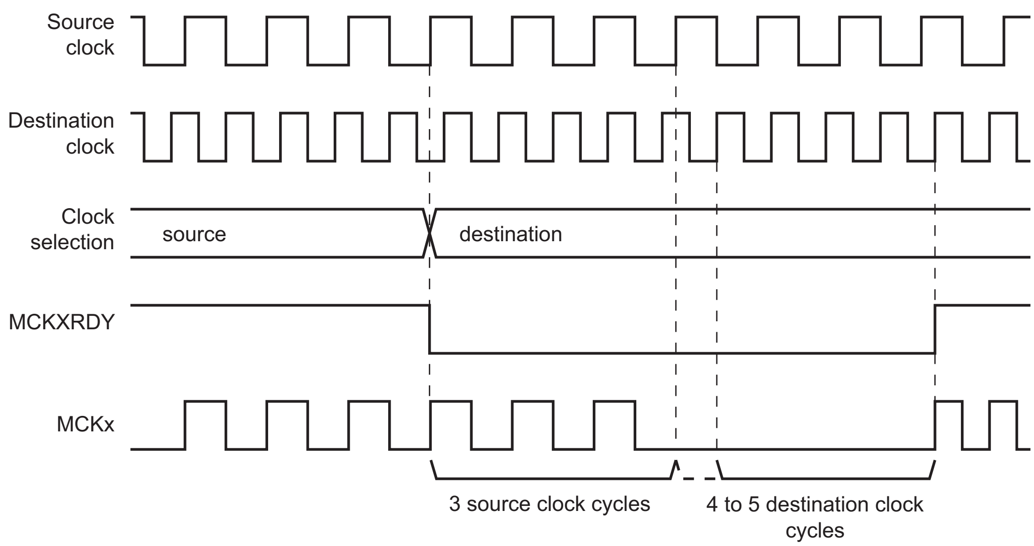


35.15.2 Main System Bus Clocks Switching Timings

The glitch-free clock switcher implemented to control the sources of MCKx (except MCK0) performs the clock switching in 3 clock cycles of the currently used clock plus 3 cycles of the target clock.

The clock switching is effective once MCKXRDY rises. See the following figure.

Figure 35-8. Switch Domain Clock (MCKx) from Source Clock to Destination Clock



35.16 Register Write Protection

To prevent any single software error from corrupting PMC behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the PMC Write Protection Mode Register (PMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PMC Write Protection Status Register (PMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PMC_WPSR.

The following registers are write-protected when the WPEN bit is set in PMC_WPMR:

- PMC System Clock Enable Register
- PMC System Clock Disable Register
- PMC PLL Control Register 0
- PMC PLL Control Register 1
- PMC PLL Spread Spectrum Register
- PMC PLL Analog Control Register
- PMC PLL Update Register
- PMC Clock Generator Main Oscillator Register
- PMC Clock Generator Main Clock Frequency Register
- PMC CPU Clock Register
- PMC CPU Clock Ratio Register
- PMC Main System Bus Clock Register
- PMC Programmable Clock Register
- PMC Fast Start-Up Mode Register
- PMC Wake-Up Control Register
- PMC Peripheral Control Register
- PMC Asynchronous Partial Wake-Up Control Register
- PMC MCK0 Monitor Limits Register

The following interrupt registers are write-protected when the WPITEN bit is set in PMC_WPMR:

- PMC Interrupt Enable Register
- PMC Interrupt Disable Register
- PMC PLL Interrupt Enable Register
- PMC PLL Interrupt Disable Register

35.17 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PMC_SCER	31:24								
		23:16								
		15:8	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
		7:0								
0x04	PMC_SCDR	31:24								
		23:16								
		15:8	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
		7:0								
0x08	PMC_SCSR	31:24								
		23:16								
		15:8	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
		7:0								
0x0C	PMC_PLL_CTRL0	31:24	ENLOCK	ENIOPLLCK	ENPLLCK	ENPLL				
		23:16					DIVIO[7:4]			
		15:8	DIVIO[3:0]							
		7:0	DIVPMC[7:0]							
0x10	PMC_PLL_CTRL1	31:24	MUL[7:0]							
		23:16			FRACR[21:16]					
		15:8	FRACR[15:8]							
		7:0	FRACR[7:0]							
0x14	PMC_PLL_SSR	31:24				ENSPREAD				
		23:16	NSTEP[7:0]							
		15:8	STEP[15:8]							
		7:0	STEP[7:0]							
0x18	PMC_PLL_ACR	31:24			LOOP_FILTER[5:0]					
		23:16						LOCK_THR[2:0]		
		15:8					CONTROL[11:8]			
		7:0	CONTROL[7:0]							
0x1C	PMC_PLL_UPDT	31:24								
		23:16			STUPTIM[5:0]					
		15:8								UPDATE
		7:0					ID[3:0]			
0x20	CKGR_MOR	31:24		AUTOCPUSW	AUTOMAINSW	BMCKIC	BMCKRST	XT32KFME	CFDEN	MOSCSEL
		23:16	KEY[7:0]							
		15:8	MOSCXTST[7:0]							
		7:0	ULP2				MOSCRCEN	ULP1		MOSCXTEN
0x24	CKGR_MCFR	31:24								CCSS
		23:16				RCMEAS				MAINFRDY
		15:8	MAINF[15:8]							
		7:0	MAINF[7:0]							
0x28	PMC_CPU_CKR	31:24								
		23:16								
		15:8						MDIV[2:0]		
		7:0		PRES[2:0]					CSS[1:0]	
0x2C	PMC_CPU_RATIO	31:24								
		23:16								
		15:8								
		7:0					RATIO[3:0]			
0x30	PMC_MCR	31:24				EN				
		23:16				CSS[4:0]				
		15:8						DIV[2:0]		
		7:0	CMD				ID[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	PMC_XTALF	31:24								
		23:16								
		15:8								
		7:0						XTALF[2:0]		
0x38 ... 0x3F	Reserved									
0x40	PMC_PCK0	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x44	PMC_PCK1	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x48	PMC_PCK2	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x4C	PMC_PCK3	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x50	PMC_PCK4	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x54	PMC_PCK5	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x58	PMC_PCK6	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x5C	PMC_PCK7	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x60	PMC_IER	31:24						MCKXRDY	PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x64	PMC_IDR	31:24						MCKXRDY	PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x68	PMC_SR	31:24						MCKXRDY	PLL_INT	GCLKRDY
		23:16	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
		15:8	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0	OSCSELS				MCKRDY			MOSCXTS
0x6C	PMC_IMR	31:24						MCKXRDY	PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x70	PMC_FSMR	31:24							WLAN1	WLAN0
		23:16					HSDHC	USBAL	RTCAL	RTTAL
		15:8								
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	PMC_WCR	31:24								CMD
		23:16							WIP	EN
		15:8								
		7:0	WKPIONB[7:0]							
0x78	PMC_FOCR	31:24								
		23:16								
		15:8								
		7:0								FOCLR
0x7C ... 0x7F	Reserved									
0x80	PMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x84	PMC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x88	PMC_PCR	31:24	CMD		GCLKEN	EN	GCLKDIV[7:4]			
		23:16	GCLKDIV[3:0]				MCKID[3:0]			
		15:8				GCLKCSS[4:0]				
		7:0		PID[6:0]						
0x8C ... 0x8F	Reserved									
0x90	PMC_SLPWK_AIPR	31:24								
		23:16								
		15:8								
		7:0								AIP
0x94	PMC_SLPWKCR	31:24				SLPWKSR				
		23:16								ASR
		15:8				CMD				
		7:0		PID[6:0]						
0x98 ... 0x9B	Reserved									
0x9C	PMC_MCKLIM	31:24	MCK_HIGH_RES[7:0]							
		23:16	MCK_LOW_RES[7:0]							
		15:8	MCK_HIGH_IT[7:0]							
		7:0	MCK_LOW_IT[7:0]							
0xA0	PMC_CSR0	31:24		PID30		PID28	PID27		PID25	PID24
		23:16	PID23	PID22	PID21		PID19			
		15:8					PID11			
		7:0								
0xA4	PMC_CSR1	31:24	PID63	PID62	PID61	PID60		PID58	PID57	PID56
		23:16	PID55			PID52	PID51		PID49	PID48
		15:8	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
		7:0	PID39	PID38	PID37			PID34	PID33	PID32
0xA8	PMC_CSR2	31:24	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
		23:16	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
		15:8	PID79	PID78	PID77		PID75	PID74	PID73	PID72
		7:0	PID71	PID70	PID69	PID68		PID66	PID65	PID64
0xAC	PMC_CSR3	31:24								
		23:16								
		15:8						PID106	PID105	PID104
		7:0						PID98	PID97	PID96

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB0 ... 0xBF	Reserved									
0xC0	PMC_GCSR0	31:24		GPID30	GPID29			GPID26		
		23:16								
		15:8								
		7:0								
0xC4	PMC_GCSR1	31:24	GPID63	GPID62	GPID61			GPID58	GPID57	
		23:16		GPID54	GPID53	GPID52	GPID51		GPID49	GPID48
		15:8	GPID47	GPID46	GPID45	GPID44	GPID43	GPID42	GPID41	GPID40
		7:0	GPID39	GPID38					GPID33	
0xC8	PMC_GCSR2	31:24					GPID91			GPID88
		23:16			GPID85	GPID84		GPID82	GPID81	GPID80
		15:8	GPID79	GPID78			GPID75	GPID74	GPID73	GPID72
		7:0	GPID71	GPID70	GPID69	GPID68		GPID66	GPID65	GPID64
0xCC ... 0xDF	Reserved									
0xE0	PMC_PLL_IER	31:24								
		23:16		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
0xE4	PMC_PLL_IDR	31:24								
		23:16		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
0xE8	PMC_PLL_IMR	31:24								
		23:16		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
0xEC	PMC_PLL_ISR0	31:24								
		23:16		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
0xF0	PMC_PLL_ISR1	31:24								
		23:16		OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
		15:8								
		7:0		UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0

35.17.1 PMC System Clock Enable Register

Name: PMC_SCER
Offset: 0x0000
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKx Programmable Clock x Output Enable

Value	Description
0	No effect.
1	Enables the corresponding Programmable Clock output.

35.17.2 PMC System Clock Disable Register

Name: PMC_SCDR
Offset: 0x0004
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKx Programmable Clock x Output Disable

Value	Description
0	No effect.
1	Disables the corresponding Programmable Clock output.

35.17.3 PMC System Clock Status Register

Name: PMC_SCSR
Offset: 0x0008
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PCK7	PCK6	PCK5	PCK4	PCK3	PCK2	PCK1	PCK0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKx Programmable Clock x Output Status

Value	Description
0	The corresponding Programmable Clock output is disabled.
1	The corresponding Programmable Clock output is enabled.

35.17.4 PMC PLL Control Register 0

Name: PMC_PLL_CTRL0
Offset: 0x000C
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
	ENLOCK	ENIOPLLCK	ENPLLCK	ENPLL				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
					DIVIO[7:4]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIVIO[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
	DIVPMC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENLOCK Enable PLL Lock

Value	Description
0	The lock signal sent by the PLL is ignored. The PLL is considered as locked once the start-up time defined by PMC_PLL_UPDT.STUPTIM has elapsed.
1	The PLL is considered as locked once the start-up time defined by PMC_PLL_UPDT.STUPTIM has elapsed and the lock signal sent by the PLL has risen.

Bit 30 – ENIOPLLCK Enable PLL Clock for IO
This feature is available for AUDIOPLL only. If the DIVIO field is equal to 0, this bit can be used to enable or disable the PLL clock on the IO. If DIVIO is not equal to 0, the PLL clock on the IO is always active.

Value	Description
0	The clock generated by the PLL is not sent to the IO.
1	The clock generated by the PLL is sent to the IO.

Bit 29 – ENPLLCK Enable PLL Clock for PMC
This feature is available for all PLLs. If the DIVPMC field is equal to 0, this bit can be used to enable or disable the PLL clock to the PMC. If DIVPMC is not equal to 0, the PLL clock to the PMC is always active.

Value	Description
0	The clock generated by the PLL is not send to the PMC.
1	The clock generated by the PLL is sent to the PMC.

Bit 28 – ENPLL Enable PLL

Value	Description
0	The PLL is off.
1	The PLL is on.

Bits 19:12 – DIVIO[7:0] Divider for PAD
Specifies the division ratio applied to the internal PLL clock before being sent to the IO. This feature is not available for all PLLs. Refer to PLL characteristics in the section “Electrical Characteristics” for more information. This feature is only available for AUDIOPLL. The frequency is defined by the following formula:

$$f_{\mathrm{IOPLLCK}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVIO} + 1)}$$

Bits 7:0 – DIVPMC[7:0] Divider for PMC
Specifies the division ratio applied to the internal PLL clock before being sent to the PMC. The frequency is defined by the following formula:

$$f_{\mathrm{PLL\ Clock}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVPMC} + 1)}$$

35.17.5 PMC PLL Control Register 1

Name: PMC_PLL_CTRL1
Offset: 0x0010
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			FRACR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FRACR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FRACR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MUL[7:0] Multiplier Factor Value
Configures the internal clock frequency. See Divider and Phase Lock Loop Programming.

Bits 21:0 – FRACR[21:0] Fractional Loop Divider Setting

35.17.6 PMC PLL Spread Spectrum Register

Name: PMC_PLL_SSR
Offset: 0x0014
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
				ENSPREAD				
Access				R/W				
Reset				0				

Bit	23	22	21	20	19	18	17	16
	NSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	STEP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	STEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – ENSPREAD Spread Spectrum Enable

Value	Description
0	The spread spectrum is not applied to the PLL.
1	The spread spectrum is applied to the PLL.

Bits 23:16 – NSTEP[7:0] Spread Spectrum Number of Steps
Specifies how many times STEP is applied to the PLL ratio. The value of NSTEP must be equal to or greater than 1.

Bits 15:0 – STEP[15:0] Spread Spectrum Step Size
When the spread spectrum is active, this field defines the step size that will be applied the PMC_PLL_CTRL1.FRACR factor. The step is applied on the LSB of PMC_PLL_CTRL1.FRACR.

35.17.7 PMC PLL Analog Control Register

Name: PMC_PLL_ACR
Offset: 0x0018
Reset: 0x00000030
Property: Read/Write

This register must be loaded with the recommended values described in the Electrical Characteristics section.

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
			LOOP_FILTER[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
						LOCK_THR[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
					CONTROL[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONTROL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0

Bits 29:24 – LOOP_FILTER[5:0] LOOP Filter Selection

Bits 18:16 – LOCK_THR[2:0] PLL Lock Threshold Value Selection

Bits 11:0 – CONTROL[11:0] PLL CONTROL Value Selection

35.17.8 PMC PLL Update Register

Name: PMC_PLL_UPDT
Offset: 0x001C
Reset: 0x00030000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			STUPTIM[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
								UPDATE
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 21:16 – STUPTIM[5:0] Start-up Time
The start-up time is defined as a number of MD_SLCK cycles and is the same for all PLLs.
STUPTIM can be modified only if all PLLs are off.

Value	Description
0	Only the lock of the PLL is considered to know the lock status of the PLL. If the lock of the PLL is not enabled, the lock never rises.
Other values	If PMC_PLL_CTRL0.ENLOCK is low, specifies the start-up time of the PLL. If PMC_PLL_CTRL0.ENLOCK is high, specifies how long the LOCK signal of the PLL is masked before being read.

Bit 8 – UPDATE PLL Setting Update (write-only)

Value	Description
0	No effect.
1	The PLL configuration written in PMC_PLL_CTRL0 and PMC_PLL_CTRL1 are applied to the PLL defined by the last ID written in the PMC_PLL_CTRL0 register.

Bits 3:0 – ID[3:0] PLL ID
When writing a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL is impacted by written fields.
When reading a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL fields are read.

Value	Name
0	PLLA
1	UTMI PLL
2	AUDIO PLL
3	LVDS PLL
4	PLLADIV2CLK

35.17.9 PMC Clock Generator Main Oscillator Register

Name: CKGR_MOR
Offset: 0x0020
Reset: 0x00000008
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.
Note: Bit 5 is always read at 1.

Bit	31	30	29	28	27	26	25	24
		AUTOCPUSW	AUTOMAINSW	BMCKIC	BMCKRST	XT32KFME	CFDEN	MOSCSEL
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MOSCXTST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ULP2				MOSCRCEN	ULP1		MOSCXTEN
Access	W				R/W	W		R/W
Reset	0				1	0		0

Bit 30 – AUTOCPUSW Automatic Processor Clock Source Switching

Value	Description
0	A main crystal oscillator failure detection has no effect on the processor clock source selection.
1	If a main crystal oscillator failure is detected, the processor clock source selection automatically switches to the main clock.

Bit 29 – AUTOMAINSW Automatic Main Clock Source Switching

Value	Description
0	A main crystal oscillator failure detection has no effect on the main clock source selection.
1	If a main crystal oscillator failure is detected, the main clock source selection automatically switches to the main RC.

Bit 28 – BMCKIC Bad MCK0 Clock Interrupt

Value	Description
0	An MCK0 clock failure has no effect on the Interrupt Controller.
1	An MCK0 clock failure detection sends an interrupt signal to the Interrupt Controller.

Bit 27 – BMCKRST Bad MCK0 Clock Reset Enable

Value	Description
0	An MCK0 clock failure detection cannot reset the system.
1	An MCK0 clock failure detection can reset the system if the reset controller is configured accordingly.

Bit 26 – XT32KFME 32.768 kHz Crystal Oscillator Frequency Monitoring Enable

Value	Description
0	The 32.768 kHz crystal oscillator frequency monitoring is disabled.
1	The 32.768 kHz crystal oscillator frequency monitoring is enabled.

Bit 25 – CFDEN Clock Failure Detector Enable

Value	Description
0	The clock failure detector is disabled.
1	The clock failure detector is enabled.

Bit 24 – MOSCSEL Main Clock Oscillator Selection

Value	Description
0	The main RC oscillator is selected.
1	The main crystal oscillator is selected.

Bits 23:16 – KEY[7:0] Write Access Password

Value	Name	Description
0x37	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bits 15:8 – MOSCXTST[7:0] Main Crystal Oscillator Start-up Time
Specifies the number of MD_SLCK cycles multiplied by 8 for the main crystal oscillator start-up time.

Bit 7 – ULP2 ULP2 Mode Command

Value	Description
0	No effect.
1	Enables the device to enter ULP2 mode. ULP2 mode is entered when the processor WFE instruction is executed.

Bit 3 – MOSCRCEN Main RC Oscillator Enable
When MOSCRCEN is set, the MOSCRCS flag is set once the main RC oscillator start-up time is achieved.

Value	Description
0	The main RC oscillator is disabled.
1	The main RC oscillator is enabled.

Bit 2 – ULP1 ULP1 Mode Command

Value	Description
0	No effect.
1	Puts the device in ULP1 mode.

Bit 0 – MOSCXTEN Main Crystal Oscillator Enable
A crystal must be connected between XIN and XOUT or a clock signal must be provided on XIN with XOUT grounded.
When MOSCXTEN is set, the MOSCXTS flag is set once the main crystal oscillator start-up time is achieved.

Value	Description
0	The main crystal oscillator is disabled.
1	The main crystal oscillator is enabled or in bypass.

35.17.10 PMC Clock Generator Main Clock Frequency Register

Name: CKGR_MCFR
Offset: 0x0024
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								CCSS
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
				RCMEAS				MAINFRDY
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
	MAINF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAINF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CCSS Counter Clock Source Selection

Value	Description
0	The measured clock of the MAINF counter is the main RC oscillator.
1	The measured clock of the MAINF counter is the main crystal oscillator.

Bit 20 – RCMEAS RC Oscillator Frequency Measure (write-only)
The measurement is performed on the main frequency (i.e., not limited to the main RC oscillator only). If the source of MAINCK is the main crystal oscillator, the restart of measurement may not be required because of the stability of crystal oscillators.

Value	Description
0	No effect.
1	Restarts measuring of the frequency of MAINCK. MAINF carries the new frequency as soon as a low-to-high transition occurs on the MAINFRDY flag.

Bit 16 – MAINFRDY Main Clock Frequency Measure Ready
To ensure that a correct value is read on the MAINF field, the MAINFRDY flag must be read at ‘1’ then another read access must be performed on the register to get a stable value on the MAINF field.

Value	Description
0	MAINF value is not valid or the measured oscillator is disabled or a measure has just been started by means of RCMEAS.
1	The measured oscillator has been enabled previously and MAINF value is available.

Bits 15:0 – MAINF[15:0] Main Clock Frequency
Gives the number of cycles of the clock selected by the bit CCSS within 16 MD_SLCK periods. To calculate the frequency of the measured clock:
f_{SELCLK} = (MAINF x f_{MD_SLCK})/16
where frequency is in MHz.

35.17.11 PMC CPU Clock Register

Name: PMC_CPU_CKR
Offset: 0x0028
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

The CSS, PRES and MDIV fields cannot be modified simultaneously. If more than one field modification is required, proceed sequentially: modify the first field and wait for PMC_SR.MCKRDY low, then modify the second field and wait for PMC_SR.MCKRDY low, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						MDIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
		PRES[2:0]					CSS[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	1

Bits 10:8 – MDIV[2:0] MCK0 Division

Value	Name	Description
0	EQ_PCK	MCK0 is FCLK divided by 1.
1	PCK_DIV2	MCK0 is FCLK divided by 2.
2	PCK_DIV4	MCK0 is FCLK divided by 4.
3	PCK_DIV3	MCK0 is FCLK divided by 3.
4	PCK_DIV5	MCK0 is FCLK divided by 5.

Bits 6:4 – PRES[2:0] Processor Clock Prescaler

Value	Name	Description
0	CLK_1	Selected clock
1	CLK_2	Selected clock divided by 2
2	CLK_4	Selected clock divided by 4
3	CLK_8	Selected clock divided by 8
4	CLK_16	Selected clock divided by 16
5	CLK_32	Selected clock divided by 32
6	CLK_64	Selected clock divided by 64
7	CLK_3	Selected clock divided by 3

Bits 1:0 – CSS[1:0] MCK0 Source Selection

Value	Name	Description
0	SLOW_CLK	MD_SLCK is selected
1	MAIN_CLK	MAINCK is selected
2	CPUPLLCK	CPUPLLCK is selected

Value	Name	Description
3	SYSPLLCK	SYSPLLCK is selected

35.17.12 PMC CPU Clock Ratio Register

Name: PMC_CPU_RATIO
Offset: 0x002C
Reset: 0x0000000F
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					RATIO[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 3:0 – RATIO[3:0] CPU Clock Ratio
Defines the ratio applied to the CPU clock.
Ratio value is (CPU_CK_RATIO+1)/16.

35.17.13 PMC Main System Bus Clock Register

Name: PMC_MCR
Offset: 0x0030
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				EN				
Access				R/W				
Reset				0				

Bit	23	22	21	20	19	18	17	16
				CSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CMD				ID[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 28 – EN Main System Bus Clock Enable

Value	Description
0	The corresponding MCKx is disabled.
1	The corresponding MCKx is enabled.

Bits 20:16 – CSS[4:0] Clock Source Selection
Select the clock to be applied to the selected clock to generate the corresponding MCKx.

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected
1	TD_SLOW_CLOCK	TD_SLCK is selected
2	MAINCK	MAINCK is selected
3	MCK0	MCK0 is selected
4	–	Reserved
5	SYSPLL	SYSPLL is selected
6	DDRPLL	DDRPLL is selected
7	IMGPLL	IMGPLL is selected
8	BAUDPLL	BAUDPLL is selected
9	AUDIOPLL	AUDIOPLL is selected
10	ETHPLL	ETHPLL is selected

Bits 10:8 – DIV[2:0] Divisor Value
Select the division ratio to be applied to the selected clock to generate the corresponding MCKx.

Value	Name	Description
0	MASTER_DIV1	Selected clock divided by 1
1	MASTER_DIV2	Selected clock divided by 2
2	MASTER_DIV4	Selected clock divided by 4
3	MASTER_DIV8	Selected clock divided by 8
4	MASTER_DIV16	Selected clock divided by 16

Value	Name	Description
5	MASTER_DIV32	Selected clock divided by 32
6	MASTER_DIV64	Selected clock divided by 64
7	MASTER_DIV3	Selected clock divided by 3

Bit 7 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bits 3:0 – ID[3:0] Main System Bus Clock Index
Main System Bus clock index selection from 1 to 4.

35.17.14 PMC Main XTAL Frequency Register

Name: PMC_XTALF
Offset: 0x0034
Reset: 0x00000007
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						XTALF[2:0]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bits 2:0 – XTALF[2:0] Crystal Frequency
Indicates the currently used main crystal frequency to UTMI PHYs. This field must match the frequency of the implemented onboard crystal frequency.

Value	Name	Description
0	F16M	The main crystal frequency is 16 MHz.
1	–	Reserved
2	F20M	The main crystal frequency is 20 MHz.
3	F24M	The main crystal frequency is 24 MHz.
4	–	Reserved
5	F32M	The main crystal frequency is 32 MHz.
6	–	Reserved
7	–	Reserved

35.17.15 PMC Programmable Clock Register

Name: PMC_PCKx
Offset: 0x40 + x*0x04 [x=0..7]
Reset: 0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				CSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 15:8 – PRES[7:0] Programmable Clock Prescaler

Value	Description
0-255	Selected clock is divided by PRES+1.

Bits 4:0 – CSS[4:0] Programmable Clock Source Selection
Values not listed are considered “reserved”.

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected.
1	TD_SLOW_CLOCK	TD_SLCK is selected.
2	MAINCK	MAINCK is selected.
3	MCK0	MCK0 is selected.
4	–	Reserved
5	SYSPLL	SYSPLL is selected.
6	DDRPLL	DDRPLL is selected.
7	IMGPLL	IMGPLL is selected.
8	BAUDPLL	BAUDPLL is selected.
9	AUDIOPLL	AUDIOPLL is selected.
10	ETHPLL	ETHPLL is selected.

35.17.16 PMC Interrupt Enable Register

Name: PMC_IER
Offset: 0x0060
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
						MCKXRDY	PLL_INT	
Access						W	W	
Reset						–	–	

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–

Bit	15	14	13	12	11	10	9	8
	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 26 – MCKXRDY Main System Bus Clock x [x=1..4] Ready Interrupt Enable

Bit 25 – PLL_INT PLL Interrupt Enable

Bit 23 – MCKMON Main System Bus Clock 0 Clock Monitor Interrupt Enable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Enable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Enable

Bit 17 – MOSCRCS Main RC Oscillator Status Interrupt Enable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Enable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKRDYx Programmable Clock Ready x Interrupt Enable

Bit 3 – MCKRDY Main System Bus Clock 0 Ready Interrupt Enable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Enable

35.17.17 PMC Interrupt Disable Register

Name: PMC_IDR
Offset: 0x0064
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
						MCKXRDY	PLL_INT	
Access						W	W	
Reset						–	–	
Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–
Bit	15	14	13	12	11	10	9	8
	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 26 – MCKXRDY Main System Bus Clock x [x=1..4] Ready Interrupt Enable

Bit 25 – PLL_INT PLL Interrupt Disable

Bit 23 – MCKMON Main System Bus Clock 0 Clock Monitor Interrupt Disable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Disable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Disable

Bit 17 – MOSCRCS Main RC Status Interrupt Disable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Disable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKRDYx Programmable Clock Ready x Interrupt Disable

Bit 3 – MCKRDY Main System Bus Clock 0 Ready Interrupt Disable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Disable

35.17.18 PMC Status Register

Name: PMC_SR
Offset: 0x0068
Reset: 0x00030008
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						MCKXRDY	PLL_INT	GCLKRDY
Access						R	R	R
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OSCSELS				MCKRDY			MOSCXTS
Access	R				R			R
Reset	0				1			0

Bit 26 – MCKXRDY Main System Bus Clock x [x=1..4] Ready Status

Value	Description
0	At least one main system bus clock is not established.
1	All main system bus clocks are established.

Bit 25 – PLL_INT PLL Interrupt Status

Value	Description
0	No PLL interrupt has occurred.
1	A PLL interrupt has occurred. PLL interrupt is defined by the configuration of PMC_IMR.

Bit 24 – GCLKRDY GCLK Ready

Value	Description
0	A GCLK is not ready to use (clock switching in progress).
1	All GCLKs are switched to their selected source clock and ready to use.

Bit 23 – MCKMON Main System Bus Clock0 Clock Monitor Error
This status is cleared on read.

Value	Description
0	Main system bus clock0 is correct or the CPU clock monitor is disabled.
1	Main system bus clock0 is incorrect or has been incorrect for an elapsed period of time since the monitoring has been enabled.

Bit 21 – XT32KERR Slow Crystal Oscillator Error

Value	Description
0	The frequency of the 32.768 kHz crystal oscillator is correct (32.768 kHz ±1%) or the monitoring is disabled.
1	The frequency of the 32.768 kHz crystal oscillator is incorrect or has been incorrect for an elapsed period of time since the monitoring has been enabled.

Bit 20 – FOS Clock Failure Detector Fault Output Status

Value	Description
0	The fault output of the clock failure detector is inactive.
1	The fault output of the clock failure detector is active. This status is cleared by writing a ‘1’ to FOCLR in PMC_FOCR.

Bit 19 – CFDS Clock Failure Detector Status

Value	Description
0	A clock failure of the main crystal oscillator clock is not detected.
1	A clock failure of the main crystal oscillator clock is detected.

Bit 18 – CFDEV Clock Failure Detector Event

Value	Description
0	No clock failure detection of the main crystal oscillator clock has occurred since the last read of PMC_SR.
1	At least one clock failure detection of the main crystal oscillator clock has occurred since the last read of PMC_SR.

Bit 17 – MOSCRCS Main RC Oscillator Status

Value	Description
0	Main RC oscillator is not stabilized.
1	Main RC oscillator is stabilized.

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status

Value	Description
0	Selection is in progress.
1	Selection is done.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKRDYx Programmable Clock Ready Status

Value	Description
0	Programmable clock x is not ready.
1	Programmable clock x is ready.

Bit 7 – OSCSELS Timing Domain Slow Clock Oscillator Selection

Value	Description
0	Embedded slow RC oscillator is selected.
1	32.768 kHz crystal oscillator is selected.

Bit 3 – MCKRDY Main System Bus Clock 0 Status

Value	Description
0	Main system bus clock 0 is not ready.
1	Main system bus clock 0 is ready.

Bit 0 – MOSCXTS Main Crystal Oscillator Status

Value	Description
0	Main crystal oscillator is not stabilized.
1	Main crystal oscillator is stabilized.

35.17.19 PMC Interrupt Mask Register

Name: PMC_IMR
Offset: 0x006C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
						MCKXRDY	PLL_INT	
Access						R	W	
Reset						0	0	

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	R		R			R	R	R
Reset	0		0			0	0	0

Bit	15	14	13	12	11	10	9	8
	PCKRDY7	PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					R			R
Reset					0			0

Bit 26 – MCKXRDY Main System Bus Clock x [x=1..4] Ready Interrupt Mask

Bit 25 – PLL_INT PLL Interrupt Mask

Bit 23 – MCKMON Main System Bus Clock 0 Monitor Error Interrupt Mask

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Mask

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Mask

Bit 17 – MOSCRCS Main RC Status Interrupt Mask

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Mask

Bits 8, 9, 10, 11, 12, 13, 14, 15 – PCKRDYx Programmable Clock Ready x Interrupt Mask

Bit 3 – MCKRDY Main System Bus Clock 0 Ready Interrupt Mask

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Mask

35.17.20 PMC Fast Start-Up Mode Register

Name: PMC_FSMR
Offset: 0x0070
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
							WLAN1	WLAN0
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
					HSDHC	USBAL	RTCAL	RTTAL
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 24, 25 – WLANx Wake-Up on LAN[x]

Value	Description
0	The Wake-up on LAN[x] alarm has no effect on the PMC.
1	The Wake-up on LAN[x] alarm enables a fast restart signal to the PMC.

Bit 19 – HSDHC HSDHC Alarm

Value	Description
0	The HSDHC alarm has no effect on the PMC.
1	The HSDHC alarm enables a fast restart signal to the PMC.

Bit 18 – USBAL USB Alarm Enable

Value	Description
0	The USB alarm has no effect on the PMC.
1	The USB alarm enables a fast restart signal to the PMC.

Bit 17 – RTCAL RTC Alarm Enable

Value	Description
0	The RTC alarm has no effect on the PMC.
1	The RTC alarm enables a fast restart signal to the PMC.

Bit 16 – RTTAL RTT Alarm Enable

Value	Description
0	The RTT alarm has no effect on the PMC.
1	The RTT alarm enables a fast restart signal to the PMC.

35.17.21 PMC Wake-Up Control Register

Name: PMC_WCR
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								CMD
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
							WIP	EN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKPIONB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 17 – WIP Wake-Up Input Polarity
Defines the active polarity of the selected wake-up input. If the corresponding wake-up input is enabled at the FSTP level, it enables a fast restart signal.

Value	Description
0	Active polarity is low.
1	Active polarity is high.

Bit 16 – EN Wake-Up Input Enable

Value	Description
0	The selected wake-up input has no effect on the PMC.
1	The selected wake-up input enables a fast restart signal to the PMC.

Bits 7:0 – WKPIONB[7:0] Wake-Up Input Number
Defines which wake-up source is to be modified during a write access (CMD is set to '1') or which wake-up source status is read on the next read access to this register (CMD is set to '0').

Primary Signal Name	WKPIONB
PAn	n
PBn	n + 32
PCn	n + 64
PDn	n + 96
PEn	n + 128

35.17.22 PMC Fault Output Clear Register

Name: PMC_FOCR
Offset: 0x0078
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FOCLR
Access								W
Reset								–

Bit 0 – FOCLR Fault Output Clear
Clears the clock failure detector fault output.

35.17.23 PMC Write Protection Mode Register

Name: PMC_WPMR
Offset: 0x0080
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

35.17.24 PMC Write Protection Status Register

Name: PMC_WPSR
Offset: 0x0084
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PMC_WPSR.
1	A write protection violation has occurred since the last read of the PMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

35.17.25 PMC Peripheral Control Register

Name: PMC_PCR
Offset: 0x0088
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CMD		GCLKEN	EN	GCLKDIV[7:4]			
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GCLKDIV[3:0]				MCKID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				GCLKCSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		PID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 31 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 29 – GCLKEN Generic Clock Enable

Value	Description
0	The selected generic clock is disabled.
1	The selected generic clock is enabled.

Bit 28 – EN Enable

Value	Description
0	The selected peripheral clock is disabled.
1	The selected peripheral clock is enabled.

Bits 27:20 – GCLKDIV[7:0] Generic Clock Division Ratio
Generic clock is the selected clock period divided by GCLKDIV + 1.
GCLKDIV must not be changed while the peripheral selects GCLKx (bit rate, etc.).

Bits 19:16 – MCKID[3:0] Main System Bus Clock Index (Read-only)
Main system bus clock index of the selected peripheral. Each peripheral is associated to one main system bus clock. It is not possible to modify this value.

Bits 12:8 – GCLKCSS[4:0] Generic Clock Source Selection

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected.
1	TD_SLOW_CLOCK	TD_SLCK is selected.
2	MAINCK	MAINCK is selected.
3	MCK0	MCK0 is selected.

Value	Name	Description
4	–	Reserved
5	SYSPLL	SYSPLL is selected.
6	DDRPLL	DDRPLL is selected.
7	IMGPLL	IMGPLL is selected.
8	BAUDPLL	BAUDPLL is selected.
9	AUDIOPLL	AUDIOPLL is selected.
10	ETHPLL	ETHPLL is selected.

Bits 6:0 – PID[6:0] Peripheral ID
Peripheral ID selection.
Not all GCLK inputs are available on all peripherals.
Refer to identifier definitions in the table “Peripheral Identifiers”.

35.17.26 PMC Asynchronous Partial Wake-Up Activity In Progress Register

Name: PMC_SLPWK_AIPR
Offset: 0x0090
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								AIP
Access								R
Reset								–

Bit 0 – AIP Activity In Progress
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOMx and ADCC.

Value	Description
0	There is no activity on peripherals. The asynchronous partial wake-up function can be activated on one or more peripherals. The device can enter ULP1 or ULP2 mode.
1	One or more peripherals are currently active. The device must not enter ULP1 or ULP2 mode. if the asynchronous partial wake-up is enabled for one of the following PIDs: FLEXCOMx and ADCC.

35.17.27 PMC Asynchronous Partial Wake-Up Control Register

Name: PMC_SLPWKCR
Offset: 0x0094
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
				SLPWKSR				
Access				R/W				
Reset				0				
Bit	23	22	21	20	19	18	17	16
								ASR
Access								R/W
Reset								0
Bit	15	14	13	12	11	10	9	8
				CMD				
Access				R/W				
Reset				0				
Bit	7	6	5	4	3	2	1	0
		PID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 28 – SLPWKSR Asynchronous Partial Wake-Up Sleep Register
Not all PIDs can be configured with asynchronous partial wake-up.
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOMx and ADCC.

Value	Description
0	The asynchronous partial wake-up function of the peripheral is disabled.
1	The asynchronous partial wake-up function of the peripheral is enabled.

Bit 16 – ASR Activity Status Register
Not all PIDs can be configured with asynchronous partial wake-up.
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOMx and ADCC.

Value	Description
0	The peripheral x is not currently active; the asynchronous partial wake-up function can be activated.
1	The peripheral x is currently active; the asynchronous partial wake-up function must not be activated.

Bit 12 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bits 6:0 – PID[6:0] Peripheral ID
Peripheral ID selection from PID2 to the maximum PID number. This refers to identifiers as defined in the table "Peripheral Identifiers".

35.17.28 PMC MCK0 Monitor Limits Register

Name: PMC_MCKLIM
Offset: 0x009C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MCK_HIGH_RES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MCK_LOW_RES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MCK_HIGH_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MCK_LOW_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MCK_HIGH_RES[7:0] MCK0 Monitoring High Reset Limit
Beyond this limit, the MCK0 frequency monitor generates a reset.

Bits 23:16 – MCK_LOW_RES[7:0] MCK0 Monitoring Low RESET Limit
Below this limit, the MCK0 frequency monitor generates a reset.

Bits 15:8 – MCK_HIGH_IT[7:0] MCK0 Monitoring High IT Limit
Beyond this limit, the MCK0 frequency monitor generates an interrupt.

Bits 7:0 – MCK_LOW_IT[7:0] MCK0 Monitoring Low IT Limit
Below this limit, the MCK0 frequency monitor generates an interrupt.

35.17.29 PMC Peripheral Clock Status Register 0

Name: PMC_CSR0
Offset: 0x00A0
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the table "Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
		PID30		PID28	PID27		PID25	PID24
Access		R		R	R		R	R
Reset		0		0	0		0	0

Bit	23	22	21	20	19	18	17	16
	PID23	PID22	PID21		PID19			
Access	R	R	R		R			
Reset	0	0	0		0			

Bit	15	14	13	12	11	10	9	8
					PID11			
Access					R			
Reset					0			

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 30 – PIDx Peripheral Clock x Status

Bits 27, 28 – PIDx Peripheral Clock x Status

Bits 21, 22, 23, 24, 25 – PIDx Peripheral Clock x Status

Bit 19 – PIDx Peripheral Clock x Status

Bit 11 – PIDx Peripheral Clock x Status

35.17.30 PMC Peripheral Clock Status Register 1

Name: PMC_CSR1
Offset: 0x00A4
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the table “Peripheral Identifiers”.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
	PID63	PID62	PID61	PID60		PID58	PID57	PID56
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	PID55			PID52	PID51		PID49	PID48
Access	R			R	R		R	R
Reset	0			0	0		0	0

Bit	15	14	13	12	11	10	9	8
	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID39	PID38	PID37			PID34	PID33	PID32
Access	R	R	R			R	R	R
Reset	0	0	0			0	0	0

Bits 28, 29, 30, 31 – PIDx Peripheral Clock x Status

Bits 23, 24, 25, 26 – PIDx Peripheral Clock x Status

Bits 19, 20 – PIDx Peripheral Clock x Status

Bits 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 – PIDx Peripheral Clock x Status

Bits 0, 1, 2 – PIDx Peripheral Clock x Status

35.17.31 PMC Peripheral Clock Status Register 2

Name: PMC_CSR2
Offset: 0x00A8
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the table "Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID79	PID78	PID77		PID75	PID74	PID73	PID72
Access	R	R	R		R	R	R	R
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID71	PID70	PID69	PID68		PID66	PID65	PID64
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bits 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Peripheral Clock x Status

Bits 4, 5, 6, 7, 8, 9, 10, 11 – PIDx Peripheral Clock x Status

Bits 0, 1, 2 – PIDx Peripheral Clock x Status

35.17.32 PMC Peripheral Clock Status Register 3

Name: PMC_CSR3
Offset: 0x00AC
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the table “Peripheral Identifiers”.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						PID106	PID105	PID104
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						PID98	PID97	PID96
Access						R	R	R
Reset						0	0	0

Bits 8, 9, 10 – PIDx Peripheral Clock x Status

Bits 0, 1, 2 – PIDx Peripheral Clock x Status

35.17.33 PMC Generic Clock Status Register 0

Name: PMC_GCSR0
Offset: 0x00C0
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the table “Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding generic clock is disabled.

1: The corresponding generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
		GPID30	GPID29			GPID26		
Access		R	R			R		
Reset		0	0			0		
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 29, 30 – GPIDx Generic Clock x Status

Bit 26 – GPIDx Generic Clock x Status

35.17.34 PMC Generic Clock Status Register 1

Name: PMC_GCSR1
Offset: 0x00C4
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the table “Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding generic clock is disabled.

1: The corresponding generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
	GPID63	GPID62	GPID61			GPID58	GPID57	
Access	R	R	R			R	R	
Reset	0	0	0			0	0	

Bit	23	22	21	20	19	18	17	16
		GPID54	GPID53	GPID52	GPID51		GPID49	GPID48
Access		R	R	R	R		R	R
Reset		0	0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	GPID47	GPID46	GPID45	GPID44	GPID43	GPID42	GPID41	GPID40
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPID39	GPID38					GPID33	
Access	R	R					R	
Reset	0	0					0	

Bits 29, 30, 31 – GPIDx Generic Clock x Status

Bits 25, 26 – GPIDx Generic Clock x Status

Bits 19, 20, 21, 22 – GPIDx Generic Clock x Status

Bits 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 – GPIDx Generic Clock x Status

Bit 1 – GPIDx Generic Clock x Status

35.17.35 PMC Generic Clock Status Register 2

Name: PMC_GCSR2
Offset: 0x00C8
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the table “Peripheral Identifiers”.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding generic clock is disabled.

1: The corresponding generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
					GPID91			GPID88
Access					R			R
Reset					0			0

Bit	23	22	21	20	19	18	17	16
			GPID85	GPID84		GPID82	GPID81	GPID80
Access			R	R		R	R	R
Reset			0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	GPID79	GPID78			GPID75	GPID74	GPID73	GPID72
Access	R	R			R	R	R	R
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPID71	GPID70	GPID69	GPID68		GPID66	GPID65	GPID64
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 27 – GPIDx Generic Clock x Status

Bit 24 – GPIDx Generic Clock x Status

Bits 20, 21 – GPIDx Generic Clock x Status

Bits 14, 15, 16, 17, 18 – GPIDx Generic Clock x Status

Bits 4, 5, 6, 7, 8, 9, 10, 11 – GPIDx Generic Clock x Status

Bits 0, 1, 2 – GPIDx Generic Clock x Status

35.17.36 PMC PLL Interrupt Enable Register

Name: PMC_PLL_IER
Offset: 0x00E0
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bits 16, 17, 18, 19, 20, 21, 22 – UNLOCKx PLL of Index x Unlock Interrupt Enable

Bits 0, 1, 2, 3, 4, 5, 6 – LOCKx PLL of Index x Lock Interrupt Enable

35.17.37 PMC PLL Interrupt Disable Register

Name: PMC_PLL_IDR
Offset: 0x00E4
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bits 16, 17, 18, 19, 20, 21, 22 – UNLOCKx PLL of Index x Unlock Interrupt Disable

Bits 0, 1, 2, 3, 4, 5, 6 – LOCKx PLL of Index x Lock Interrupt Disable

35.17.38 PMC PLL Interrupt Mask Register

Name: PMC_PLL_IMR
Offset: 0x00E8
Reset: 0x00000000
Property: Read-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22 – UNLOCKx PLL of Index x Unlock Interrupt Mask

Bits 0, 1, 2, 3, 4, 5, 6 – LOCKx PLL of Index x Lock Interrupt Mask

35.17.39 PMC PLL Interrupt Status Register 0

Name: PMC_PLL_ISR0
Offset: 0x00EC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		UNLOCK6	UNLOCK5	UNLOCK4	UNLOCK3	UNLOCK2	UNLOCK1	UNLOCK0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LOCK6	LOCK5	LOCK4	LOCK3	LOCK2	LOCK1	LOCK0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22 – UNLOCKx PLL of Index x Unlock Interrupt Status

Value	Description
0	PLLx is not unlocked.
1	PLLx is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bits 0, 1, 2, 3, 4, 5, 6 – LOCKx PLL of Index x Lock Interrupt Status

Value	Description
0	PLLx is not locked.
1	PLLx is locked.

35.17.40 PMC PLL Interrupt Status Register 1

Name: PMC_PLL_ISR1
Offset: 0x00F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22 – OVRx PLLx Overflow

Value	Description
0	PLL is not in Overflow state.
1	PLL encountered an overflow.

Bits 0, 1, 2, 3, 4, 5, 6 – UDRx PLLx Underflow

Value	Description
0	PLL is not in Underflow state.
1	PLL encountered an underflow.

36. Parallel Input/Output Controller (PIO)

36.1 Description

The Parallel Input/Output Controller (PIO) manages up to 136 fully programmable input/output lines. Each I/O line may be dedicated as a general purpose I/O or be assigned to a function of an embedded peripheral. This ensures effective optimization of the pins of the product.

The PIO Controller features a synchronous output providing up to 32 bits of data output in a single write operation.

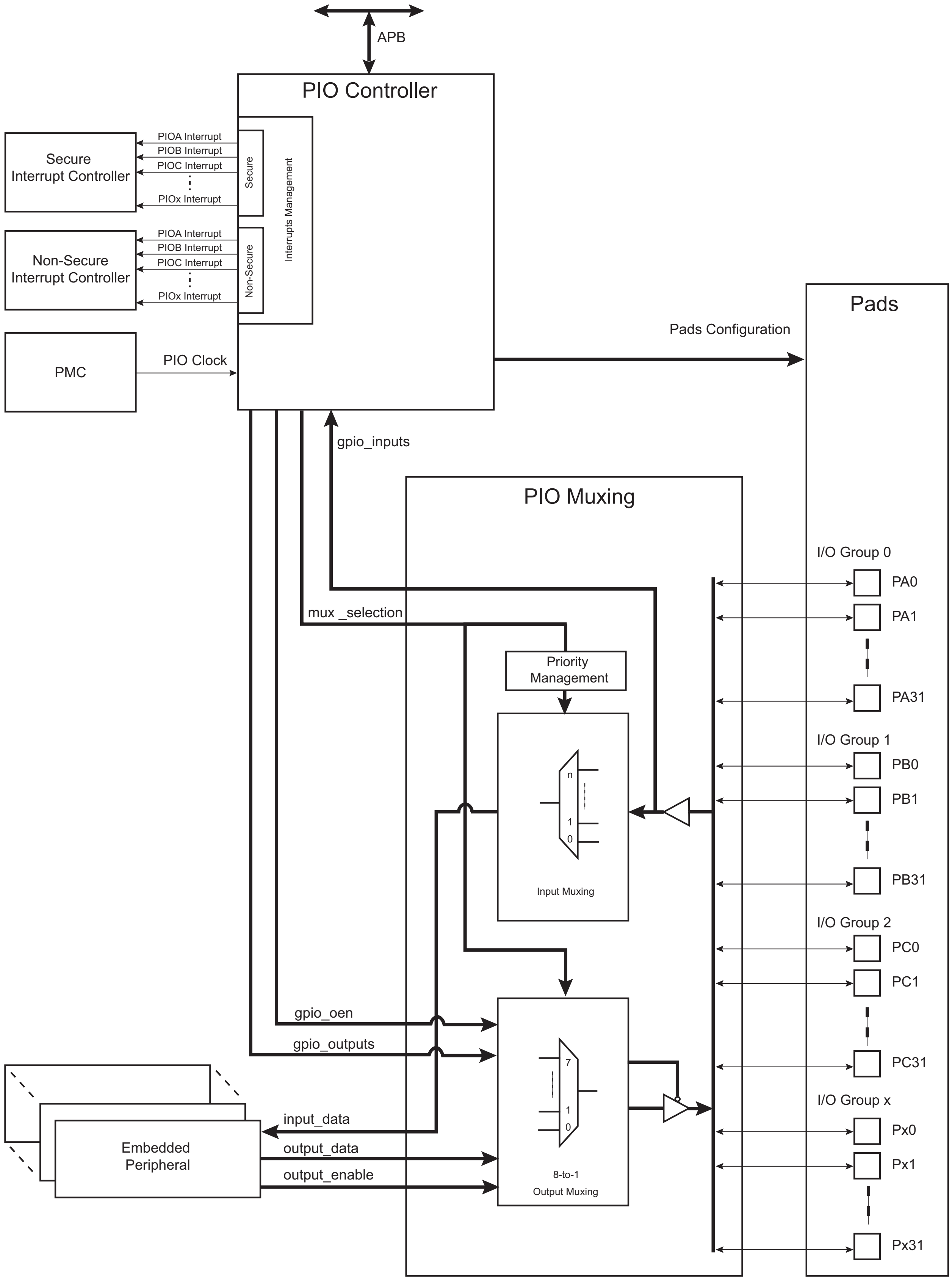
The PIO embeds safety and security features.

36.2 Embedded Characteristics

- Up to 136 Programmable I/O Lines
- Multiplexing of Up to 7 Peripheral Functions per I/O Line
- For Each I/O Line (whether assigned to a peripheral or used as general purpose I/O):
  - Input change interrupt
  - Programmable glitch filter
  - Programmable debouncing filter
  - Multi-drive option enables driving in open drain
  - Programmable pull-up/pull-down
  - Pin data status register, supplies visibility of the level on the pin at any time
  - Programmable event: rising edge, falling edge, both edges, low-level or high-level
  - Configuration lock by the connected peripheral
  - Secure or Non-Secure management
  - Programmable configuration lock (active until next V_{DDCORE} reset) to protect against further software modifications (intentional or unintentional)
  - Programmable freeze on tamper event for each I/O line configuration
- Register Write Protection against Unintentional Software Modifications:
  - One configuration bit to enable or disable protection of I/O line settings
  - One configuration bit to enable or disable protection of interrupt settings
- Synchronous Output, Possibility to Set or Clear Simultaneously Up to 32 I/O Lines in a Single Write
- Programmable Schmitt Trigger Inputs
- Programmable Slew Rate

36.3 Block Diagram

Figure 36-1. PIO Controller Block Diagram



Notes:
1. x = 4 (the number of I/O groups is 5).
2. n depends on the number of I/O lines affected to the IP input.

36.4 Product Dependencies

36.4.1 Pin Multiplexing

Each pin is configurable, depending on the product, as either a general purpose I/O line only, or as an I/O line multiplexed with up to 7 peripheral I/Os. As the multiplexing is hardware defined and thus product-dependent, the hardware designer and programmer must carefully determine the configuration of the PIO Controllers required by their application. When an I/O line is general purpose only, i.e., not multiplexed with any peripheral I/O, programming of the PIO Controller regarding the assignment to a peripheral has no effect and only the PIO Controller can control how the pin is driven by the product.

36.4.2 External Interrupt Lines

The interrupt signals IRQ0 to IRQn are multiplexed through the PIO Controllers.

36.4.3 Power Management

The Power Management Controller (PMC) controls the PIO Controller clock in order to save power. Writing any of the registers of the user interface does not require the PIO Controller clock to be enabled. This means that the configuration of the I/O lines does not require the PIO Controller clock to be enabled.

However, when the clock is disabled, not all of the features of the PIO Controller are available, including glitch filtering. Note that the input change interrupt, the interrupt modes on a programmable event and the read of the pin level require the clock to be validated.

After a hardware reset, the PIO clock is disabled by default.

The user must configure the PMC before any access to the input line information.

36.4.4 Interrupt Generation

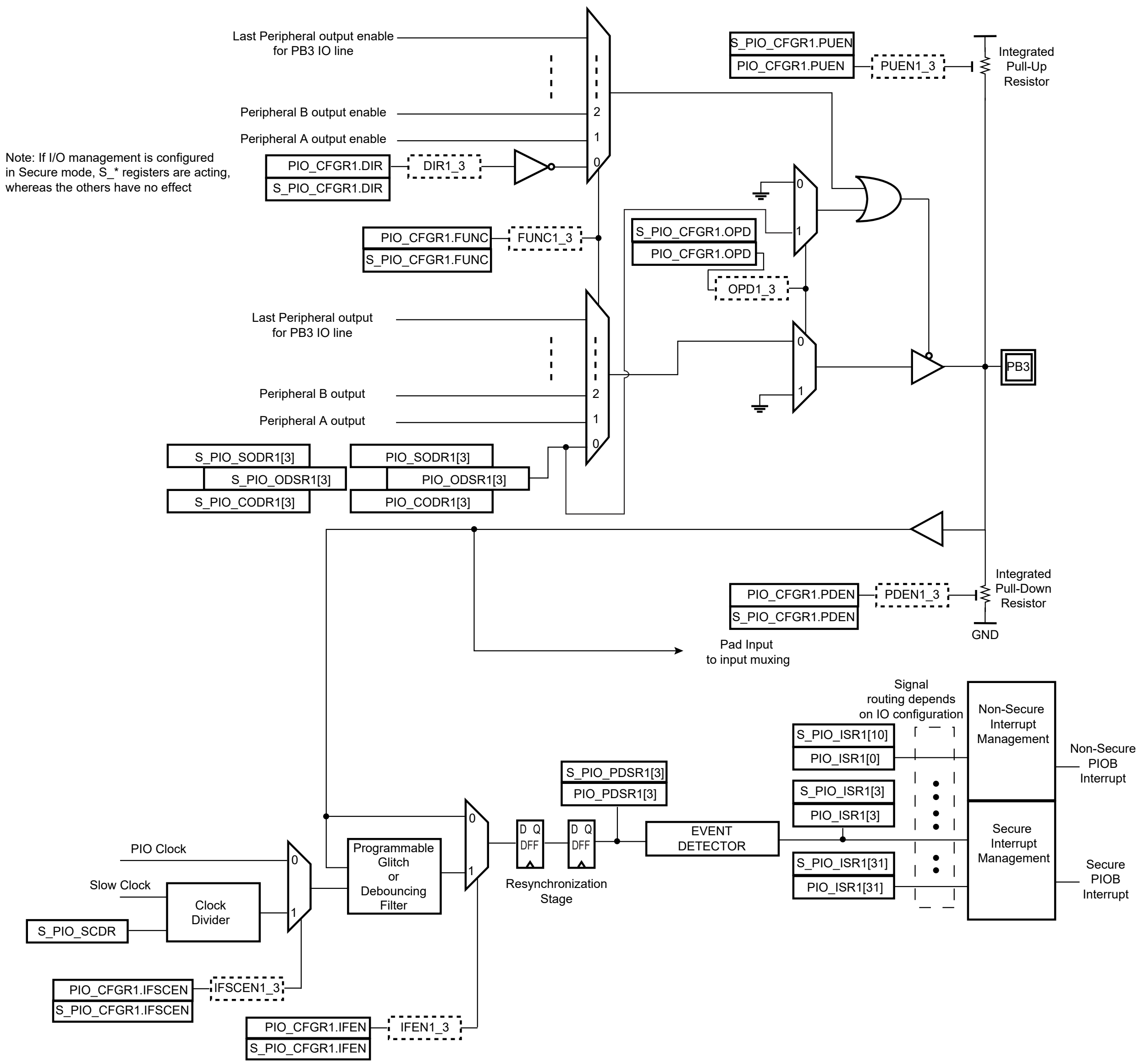
For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources. The PIO Controller supplies one interrupt signal per I/O group. Refer to the PIO Controller peripheral identifier in the product description to identify the interrupt sources dedicated to the PIO Controller. The PIO Controller can target either the Secure or Non-Secure Interrupt Controller according to security level of the I/O line which triggers the interruption. Using the PIO Controller requires the Interrupt Controller to be programmed first.

The PIO Controller interrupt can be generated only if the PIO Controller clock is enabled.

36.5 Functional Description

The PIO Controller features up to 512 fully-programmable I/O lines. Most of the control logic associated to each I/O is represented in the following figure, where the I/O line 3 of the PIOB (PB3) is described as an example. In this description each signal shown represents one of up to 512 possible indexes.

Figure 36-2. I/O Line Control Logic



36.5.1 I/O Line Configuration Method

The user interface of the PIO Controller provides several sets of registers. Each set of registers interfaces with one I/O group.

Table 36-1. I/O Group List

I/O Group Number	PIO
0	PIOA
1	PIOB
...	...
4	PIOE

36.5.1.1 Security Management

The user must first define the security level of the I/O line. Each I/O line of each I/O group must be defined as either secure or non-secure lines. Each I/O line of the I/O group x can be set as non-secure I/O line by writing a 1 to the corresponding bit P0–P31 of the Secure PIO Set I/O Non-Secure register (S_PIO_SIONRx) of the I/O group x.

To define an I/O line of I/O group x as a secure I/O line, write a 1 to the corresponding bit P0–P31 of the Secure PIO Set I/O Secure register (S_PIO_SIOSRx) of the I/O group x.

Examples:

Setting the I/O line PC4 as a non-secure line:

- Write the value 16 (bit 4 at 1) at address 0x10B0 (S_PIO_SIONR2)

Setting the I/O line PB3 as a secure line:

- Write the value 8 (bit 3 at 1) at address 0x1074 (S_PIO_SIOSR1)

The security level of each I/O line is reported by the Secure PIO I/O Security Status register (S_PIO_IOSSRx) of the corresponding I/O group. Reading 0 at the corresponding bit P0–P31 means that the corresponding I/O line of the I/O group is defined as secure. Reading 1 means that this I/O line of the I/O group is non-secure.

The PIO Controller user interface is divided into two register mapping areas:

- The Non-Secure area, located from address 0x0 to 0x1000, can be accessed by any (secure or non-secure) host. This area interfaces with all the I/O lines defined as non-secure. Trying to access to an I/O line defined as secure through this area will have no effect on the I/O line and the read values will be 0.
- The Secure area, located above address 0x1000, can only be accessed by a secure host (if the PIO Controller is defined as secure at the HMATRIX level). This area interfaces with all the I/O lines defined as secure. Trying to access to an I/O line defined as non-secure through this area will have no effect on the I/O line and the read values will be 0.

36.5.1.2 Programming I/O Line Configuration

The user must first define which I/O line in the group will be targeted by writing a 1 to the corresponding bit in the PIO Mask Register (PIO_MSKRx). Several I/O lines in an I/O group can be configured at the same time by setting the corresponding bits in PIO_MSKRx. Then, writing the PIO Configuration Register (PIO_CFGRx) apply the configuration to the I/O line(s) defined in PIO_MSKRx. All the I/O lines defined as secure in the S_PIO_SIOSRx must be configured by writing the S_PIO_CFGRx and S_PIO_MSKRx registers.

For more details concerning the I/O line configuration using PIO_MSKRx and PIO_CFGRx, see section I/O Lines Programming Example.

36.5.1.3 Reading the I/O Line Configuration

As for programming operation, reading configuration requires the user to first define which I/O line in the group x will be targeted by writing a 1 to the corresponding bit in the PIO_MSKRx. The value of the targeted I/O line is read in PIO_CFGRx.

If several bits are set in PIO_MSKRx, then the read configuration in PIO_CFGRx is the configuration of the I/O line with the lowest index.

Note that S_PIO_MSKRx and S_PIO_CFGRx must be used to read the configuration of a secure I/O line.

36.5.2 Pull-Up and Pull-Down Resistor Control

Each I/O line is designed with an embedded pull-up resistor and an embedded pull-down resistor.

The pull-up resistor on the I/O line(s) defined in PIO_MSKRx can be enabled by setting the PUEN bit in PIO_CFGRx. Clearing the PUEN bit in PIO_CFGRx disables the pull-up resistor of I/O lines defined in PIO_MSKRx.

The pull-down resistor on the I/O line(s) defined in PIO_MSKRx can be enabled by setting the PDEN bit in PIO_CFGRx. Clearing the PDEN bit in PIO_CFGRx disables the pull-down resistor of I/O lines defined in PIO_MSKRx.

If both PUEN and PDEN bits are set in PIO_CFGRx, only the pull-up resistor is enabled for I/O line(s) defined in PIO_MSKRx and the PDEN bit is discarded.

Control of the pull-up resistor is possible regardless of the configuration of the I/O line (Input, Output, Open-drain).

Note that S_PIO_MSKRx and S_PIO_CFGRx must be used to program the pull-up or pull-down configuration of a secure I/O line.

For more details concerning Pull-up and Pull-down configuration, see PIO_CFGRx or S_PIO_CFGRx for secure I/O line configuration.

The reset value of PUEN and PDEN bits of each I/O line is defined at the product level and depends on the multiplexing of the device.

36.5.3 General Purpose or Peripheral Function Selection

The PIO Controller provides multiplexing of up to 7 peripheral functions on a single pin. The selection is performed by writing the FUNC field in PIO_CFGRx. The selected function is applied to the I/O line(s) defined in PIO_MSKRx.

When FUNC is 0, no peripheral is selected and the General Purpose PIO (GPIO) mode is selected (in this mode, the I/O line is controlled by the PIO Controller).

When the value configured in PIO_CFGRx.FUNC is greater than 0, the software cannot drive the I/O line anymore and the value configured in PIO_CFGRx.FUNC defines which embedded peripheral drives the I/O lineFor more details, refer to the table Pin Description in section Package and Pinout.

Note that S_PIO_MSKRx and S_PIO_CFGRx must be used to program the FUNC field of a secure I/O line.

For more details, see PIO_CFGRx or S_PIO_CFGRx for secure I/O line configuration.

Note that multiplexing of peripheral lines affects both input and output peripheral lines. When a peripheral is not selected on any I/O line, its inputs are assigned with constant default values defined at the product level. The user must ensure that only one I/O line is affected to a peripheral input at a time.

The reset value of the FUNC field of each I/O line is defined at the product level and depends on the multiplexing of the device.

36.5.4 Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding FUNC field of the line configuration is 1, the drive of the I/O line (direction, output value) is controlled by the peripheral.

When the FUNC field of a I/O line is 0, then the I/O line is set in General Purpose mode and the I/O line can be configured to be driven by the PIO Controller (software) instead of the peripheral.

If PIO_CFGRx/S_PIO_CFGRx/.DIR is configured in Output mode and PIO_CFGRx/S_PIO_CFGRx/.FUNC=0, then the I/O line can be driven by the PIO Controller. The level driven on an I/O line can be determined by writing in the PIO Set Output Data Register (PIO_SODRx)/Secure PIO Set Output Data Register (S_PIO_SODRx)/ and the PIO Clear Output Data Register (PIO_CODRx)Secure PIO Clear Output Data Register (S_PIO_CODRx)/. These write operations, respectively, set and clear the PIO Output Data Status Register (PIO_ODSRx)/Secure PIO Output Data Status Register (S_PIO_ODSRx)/, which represents the data driven on the I/O lines. Writing PIO_ODSRx/ S_PIO_ODSRx directly is possible and only affects the I/O line set to 1 in PIO_MSKRx/S_PIO_MSKRx (see Synchronous Data Output).

When DIR of the I/O line configuration is at zero, the corresponding I/O line is used as an input only.

DIR has no effect if the corresponding line is assigned to a peripheral function, but writing DIR is managed whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO_SODRx/S_PIO_SODRx and PIO_CODRx/S_PIO_CODRx affects PIO_ODSRx/ S_PIO_ODSRx. This is important as it defines the first level driven on the I/O line.

36.5.5 Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODRx/S_PIO_SODRx and PIO_CODRx/S_PIO_CODRx. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSRx/S_PIO_ODSRx. Only I/O lines set to 1 in PIO_MSKRx/S_PIO_MSKRx are written.

36.5.6 Open-Drain Mode

Each I/O can be independently programmed in Open-Drain mode. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pull-up resistor (or enabling of the internal one) is generally required to ensure a high level on the line.

The Open-Drain mode is controlled by the OPD bit in the I/O line configuration (PIO_CFGRx or S_PIO_CFGRx). An I/O line is switched in Open-Drain mode by setting the PIO_CFGRx/S_PIO_CFGRx.OPD bit. The Open-Drain mode can be selected if the I/O line is not controlled by a peripheral (the FUNC field must be cleared in PIO_CFGRx/S_PIO_CFGRx).

For more details concerning the Open-Drain mode, see PIO_CFGRx or S_PIO_CFGRx for secure I/O line configuration.

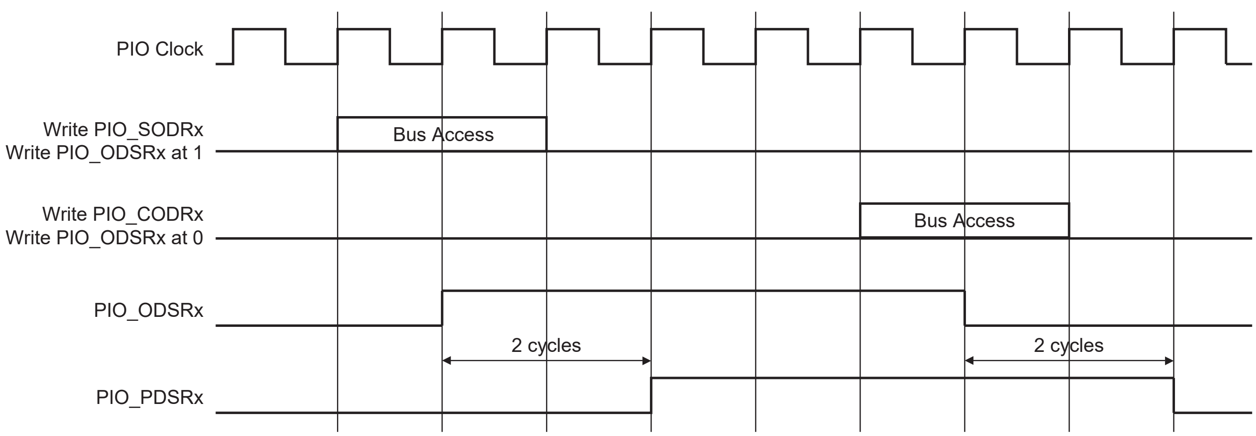
After reset, the OPD bit of each I/O line is defined at the product level and depends on the multiplexing of the device.

Note: Open-drain capability is not possible when the I/O line is driven by a peripheral. Only software control (GPIO mode) is able to manage the open-drain for an I/O line. TWI is able to manage open-drain because this peripheral does not require the PIO to be configured in Open-drain mode.

36.5.7 Output Line Timings

The figure below shows how the outputs are driven either by writing PIO_SODRx/S_PIO_SODRx or PIO_CODRx/S_PIO_CODRx, or by directly writing PIO_ODSRx/S_PIO_ODSRx. This last case is valid only if the corresponding bit in PIO_MSKRx/S_PIO_MSKRx is set. The figure also shows when the feedback in the Pin Data Status register (PIO_PDSRx/S_PIO_PDSRx) is available.

Figure 36-3. Output Line Timings



36.5.8 Inputs

The level on each I/O line of the I/O group x can be read through PIO_PDSRx/S_PIO_PDSRx. This register indicates the level of the I/O lines regardless of their configuration, whether uniquely as an input, or driven by the PIO Controller, or driven by a peripheral.

Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSRx/S_PIO_PDSRx reads the levels present on the I/O line at the time the clock was disabled.

36.5.9 Input Glitch and Debouncing Filters

Optional input glitch and debouncing filters are independently programmable on each I/O line.

The glitch filter can filter a glitch with a duration of less than 1 peripheral clock and the debouncing filter can filter a pulse of less than 1 period of a programmable divided slow clock.

The selection between glitch filtering or debounce filtering is done by writing the PIO_CFGR.IFSCEN. The selected filtering mode is applied to the I/O line(s) defined in PIO_MSKRx.

- If IFSCEN = 0: The glitch filter can filter a glitch with a duration of less than 1 peripheral clock period.
- If IFSCEN = 1: The debouncing filter can filter a pulse with a duration of less than 1 programmable divided slow clock period.

For the debouncing filter, the period of the divided slow clock is performed by writing in the DIV field of the Secure PIO Slow Clock Divider Debouncing Register (S_PIO_SCDR): $t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$.

When the glitch or debouncing filter is enabled, a glitch or pulse with a duration of less than 1 selected clock cycle (selected clock represents PIO clock or divided slow clock depending on IFSCEN configuration) is automatically rejected. A pulse of a duration equal to 1 clock cycle may be filtered or not depending on the clock jitter.

The filters also introduce some latencies, illustrated in the figures below .

The glitch filter of each I/O line is controlled by PIO_CFGR.IFEN. Setting PIO_CFGRx.IFEN enables the glitch filter of the I/O line(s) defined in PIO_MSKRx.

When the glitch and/or debouncing filter is enabled, it does not modify the behavior of the inputs on the peripherals. It acts only on the value read in PIO_PDSRx and on the input change interrupt detection. The glitch and debouncing filters require that the PIO Controller clock is enabled.

Figure 36-4. Input Glitch Filter Timing

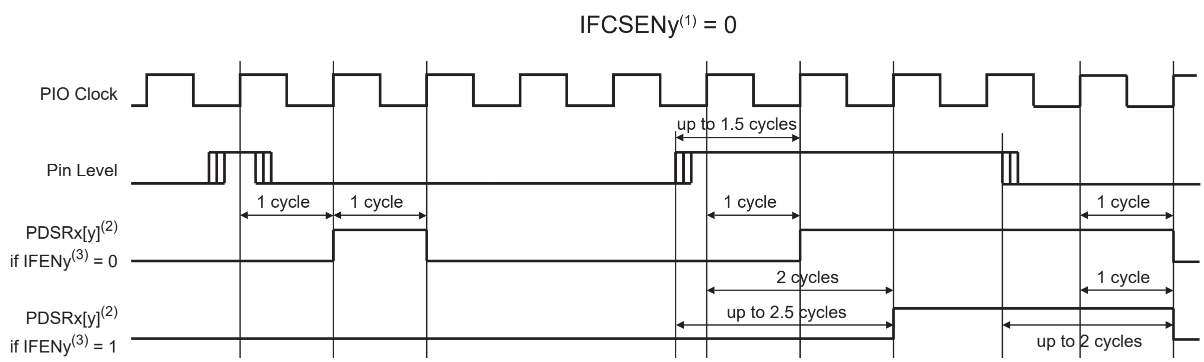
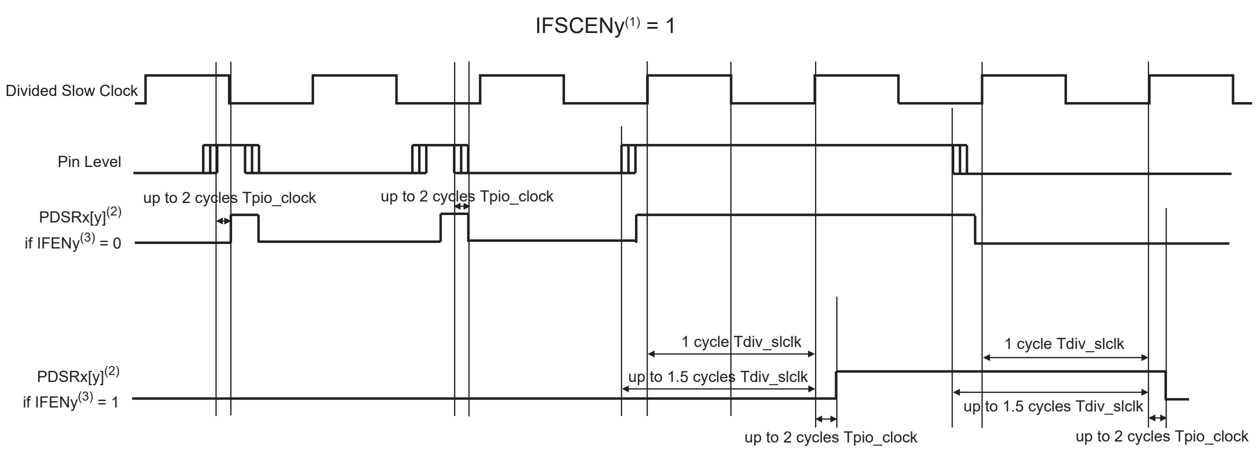


Figure 36-5. Input Debouncing Filter Timing



Note:

1. Means IFCSEN of the I/O line y of the I/O group x.
2. Means PIO Data Status value of the I/O line y of the I/O group x.
3. Means IFEN of the I/O line y of the I/O group x.

36.5.10 Input Edge/Level Interrupt

Each I/O group can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupts are controlled by writing the PIO Interrupt Enable Register (PIO_IERx) and the PIO Interrupt Disable Register (PIO_IDRx), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the PIO Interrupt Mask Register (PIO_IMRx). For the Secure I/O lines, the Input Edge/Level interrupts are controlled by writing S_PIO_IERx and S_PIO_IDRx, which enable and disable input change interrupts respectively by setting and clearing the corresponding bit in the S_PIO_IMRx. As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the PIO Controller clock must be enabled. The Input Change interrupt is available regardless of the configuration of the I/O line, i.e., configured as an input only, controlled by the PIO Controller or assigned to a peripheral function.

Each I/O group can generate a Non-Secure interrupt and a Secure interrupt according to the security level of the I/O line which triggers the interrupt.

According to the EVTSEL field value in PIO_CFGRx or S_PIO_CFGRx in case of a Secure I/O line, the interrupt signal of the I/O group x can be generated on the following occurrence:

- (S_)PIO_CFGRx.EVTSELy = 0: The interrupt signal of the I/O group x is generated on the I/O line y falling edge detection (assuming that (S_)PIO_IMRx[y] = 1).
- (S_)PIO_CFGRx.EVTSELy = 1: The interrupt signal of the I/O group x is generated on the I/O line y rising edge detection (assuming that (S_)PIO_IMRx[y] = 1).
- (S_)PIO_CFGRx.EVTSELy = 2: The interrupt signal of the I/O group x is generated on the I/O line y both rising and falling edge detection (assuming that (S_)PIO_IMRx[y] = 1).
- (S_)PIO_CFGRx.EVTSELy = 3: The interrupt signal of the I/O group x is generated on the I/O line y low level detection (assuming that (S_)PIO_IMRx[y] = 1).
- (S_)PIO_CFGRx.EVTSELy = 4: The interrupt signal of the I/O group x is generated on the I/O line y high level detection (assuming that (S_)PIO_IMRx[y] = 1).

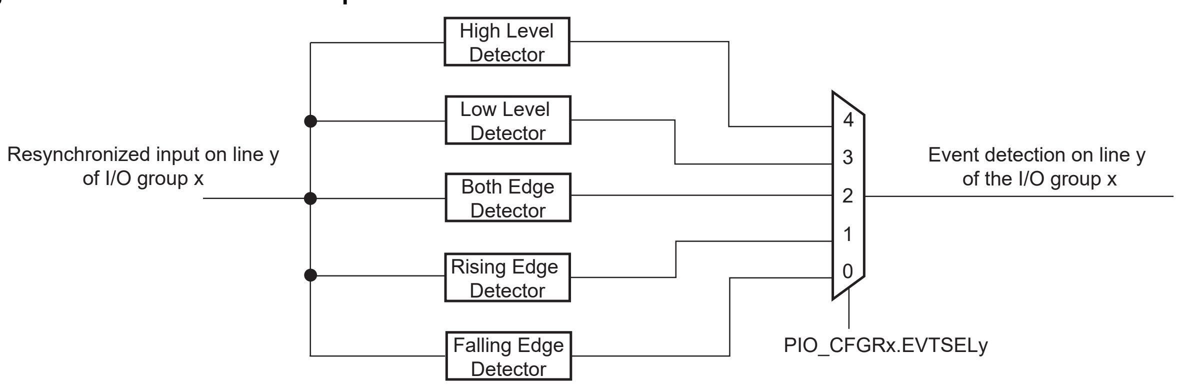
By default, the interrupt can be generated at any time a falling edge is detected on the input.

When an input edge or level is detected on an I/O line, the corresponding bit in the PIO Interrupt Status Register (PIO_ISRx), or in the Secure PIO Interrupt Status Register (S_PIO_ISRx) if the I/O line is Secure, is set.

For a Non-Secure I/O line, if the corresponding bit in PIO_IMRx is set, the Non-Secure interrupt line of the I/O group x is asserted. For a Secure I/O line, if the corresponding bit in S_PIO_IMRx is set, the Secure interrupt line of the I/O group x is asserted.

When the software reads PIO_ISRx, all the Non-Secure interrupts of the I/O group x are automatically cleared. When the software reads S_PIO_ISRx, all the Secure interrupts of the I/O group x are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISRx or S_PIO_ISRx are read must be handled. When an interrupt is enabled on a “level”, the interrupt is generated as long as the interrupt source is not cleared, even if some read accesses in PIO_ISRx or S_PIO_ISRx are performed.

Figure 36-6. Event Detector on Input Lines



Example of interrupt generation on following lines:

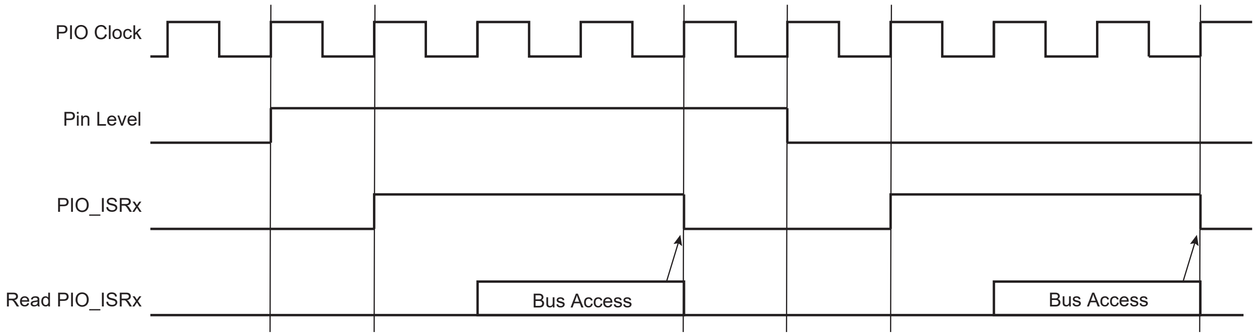
- Rising edge on the Secure PIO line 0 of the I/O group 0 (PIOA)
- Low-level edge on the Secure PIO line 1of the I/O group 0 (PIOA)
- Rising edge on the Secure PIO line 2 of the I/O group 0 (PIOA)
- High-level on the Secure PIO line 3 of the I/O group 0 (PIOA)
- Low-level on the Non-Secure PIO line 4 of the I/O group 0 (PIOA)
- High-level on the Secure PIO line 0 of the I/O group 1 (PIOB)
- Falling edge on the Secure PIO line 1 of the I/O group 1 (PIOB)
- Rising edge on the Secure PIO line 2 of the I/O group 1 (PIOB)
- Any edge on the other Non-Secure lines of the I/O group 1 (PIOB)

The table below details the required configuration for this example.

Table 36-2. Configuration for Example Interrupt Generation

Configuration	Name
PIOA: I/O Line Security Level	Define the I/O lines 0 to 3 of the PIOA as Secure by writing 32’h0000_000F in the S_PIO_SIOSR0 (offset 0x1034) Define the I/O lines 4 of the PIOA as Non-Secure by writing 32’h0000_0010 in the S_PIO_SIONR0 (offset 0x1030)
PIOA: Interrupt Mode	Enable interrupt sources for lines 0 to 3 of PIOA by writing 32’h0000_000F in S_PIO_IER0 (offset 0x1020) Enable interrupt source for the line 4 of PIOA by writing 32’h0000_0010 in PIO_IER0 (offset 0x20)
PIOA: Event Selection	Configure Rising Edge detection for Secure lines 0 and 2: Write 32’h0000_0005 in S_PIO_MSKR0 (offset 0x1000) Write 32’h0100_0000 in S_PIO_CFGR0 (offset 0x1004)
	Configure Low Level detection for Secure line 1: Write 32’h0000_0002 in S_PIO_MSKR0 (offset 0x1000) Write 32’h0300_0000 in S_PIO_CFGR0 (offset 0x1004)
	Configure High Level detection for Secure line 3: Write 32’h0000_0008 in S_PIO_MSKR0 (offset 0x1000) Write 32’h0400_0000 in S_PIO_CFGR0 (offset 0x1004)
	Configure Low Level detection for Non-Secure line 4: Write 32’h0000_0010 in PIO_MSKR0 (offset 0x0) Write 32’h0300_0000 in PIO_CFGR0 (offset 0x4)
PIOB: I/O Line Security Level	Define the I/O lines 0 to 2 of the PIOB as Secure by writing 32’h0000_0007 in the S_PIO_SIOSR1 (offset 0x1074) Define the other I/O lines of the PIOB as Non-Secure by writing 32’hFFFF_FFF8 in the S_PIO_SIONR1 (offset 0x1070)
PIOB: Interrupt Mode	Enable interrupt sources for lines 0 to 2 of PIOB by writing 32’h0000_0007 in S_PIO_IER1 (offset 0x1060) Enable interrupt sources for all other lines of PIOB by writing 32’hFFFF_FFF8 in PIO_IER1 (offset 0x60)
PIOB: Event Selection	Configure High Level detection for Secure line 0: Write 32’h0000_0001 in S_PIO_MSKR1 (offset 0x1040) Write 32’h0400_0000 in S_PIO_CFGR1 (offset 0x1044)
	Configure Falling Edge detection for Secure line 1: Write 32’h0000_0002 in S_PIO_MSKR1 (offset 0x1040) Write 32’h0000_0000 in S_PIO_CFGR1 (offset 0x1044)
	Configure Rising Edge detection for Secure line 2: Write 32’h0000_0004 in S_PIO_MSKR1 (offset 0x1040) Write 32’h0100_000 in S_PIO_CFGR1 (offset 0x1044)
	Configure Low Level detection for Non-Secure lines: Write 32’hFFFF_FFF8 in PIO_MSKR1 (offset 0x40) Write 32’h0200_000 in PIO_CFGR1 (offset 0x44)

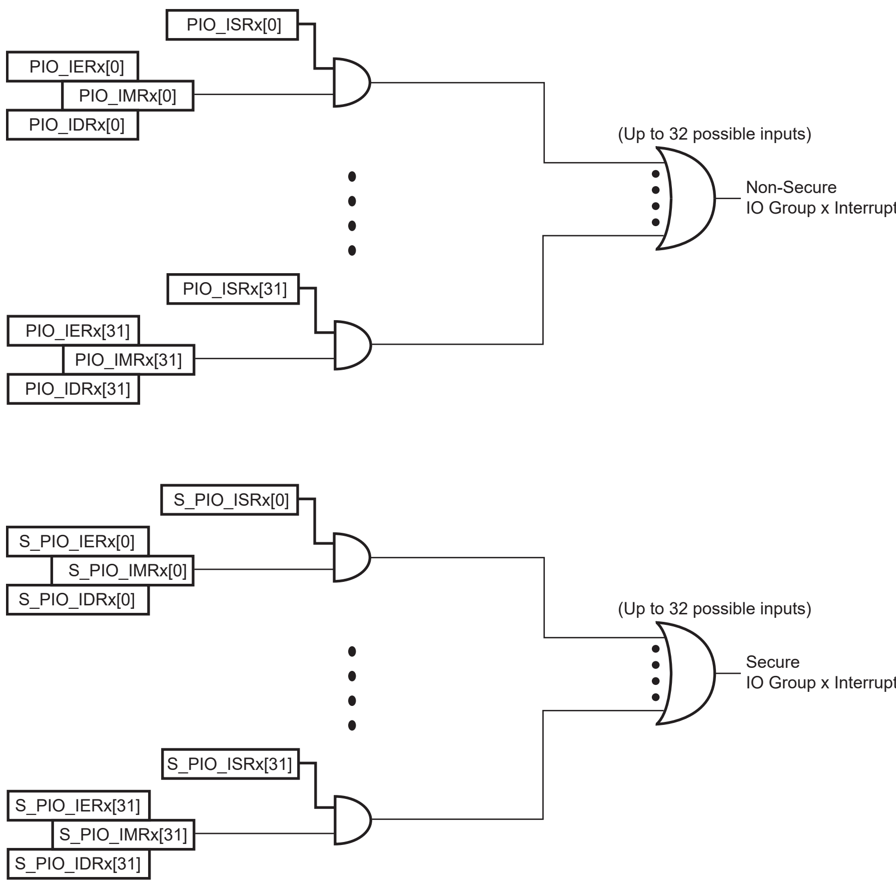
Figure 36-7. Input Change Interrupt Timings When No Additional Interrupt Modes



36.5.11 Interrupt Management

The PIO Controller can drive one secure interrupt signal and one non-secure interrupt signal per I/O group (refer to the Block Diagram). Secure interrupt signals are connected to the secure interrupt controller of the system. Non-secure interrupt signals are connected to the non-secure interrupt controller of the system.

Figure 36-8. PIO Interrupt Management



36.5.12 I/O Lines Lock

When an I/O line is controlled by a peripheral (particularly the Pulse Width Modulation Controller PWM), it can become locked by the action of this peripheral via an input of the PIO Controller. When an I/O line is locked, the following fields in PIO_CFGRx/S_PIO_CFGRx are locked and cannot be modified:

- FUNC: Peripheral selection cannot be changed when the corresponding I/O line is locked.
- PUEN: Pull-Up configuration cannot be changed when the corresponding I/O line is locked.

- PDEN: Pull-Down configuration cannot be changed when the corresponding I/O line is locked.
- OPD: Open Drain configuration cannot be changed when the corresponding I/O line is locked.

Writing to one of these fields while the corresponding I/O line is locked will have no effect.

The user can know at anytime which I/O line is locked by reading the PIO Lock Status Register (PIO_LOCKSR) or Secure PIO Lock Status Register (S_PIO_LOCKSR) for locked Secure I/O lines. Once an I/O line is locked, the only way to unlock it is to apply a hardware reset to the PIO Controller.

36.5.13 Programmable I/O Drive

It is possible to configure the I/O drive for pads PA0 to PE7. The I/O drive of the pad can be programmed by writing the DRVSTR field in the PIO_CFGRx if the corresponding line is Non-Secure or S_PIO_CFGRx if the I/O line is Secure. For details, refer to the section “Electrical Characteristics”.

Note: When SDHC drives I/O lines, PIO Drive control (PIO_CFGRx) has no effect; only the drive control from SDHC is operational

36.5.14 Programmable Schmitt Trigger

It is possible to configure each input for the Schmitt trigger. The Schmitt trigger can be enabled by setting PIO_CFGRx.SCHMITT if the corresponding line is Non-Secure or S_PIO_CFGRx if the I/O line is Secure. By default, the Schmitt trigger is active.

36.5.15 Programmable Slew Rate

Each output can be configured for slew rate control. Slew rate control can be enabled by setting the bit SR of the PIO_CFGRx if the corresponding line is Non-Secure, or S_PIO_CFGRx if the I/O line is Secure. For details, refer to the section “Electrical Characteristics”.

Slew rate control does not apply to high-speed I/O lines.

36.5.16 I/O Line Configuration Freeze

36.5.16.1 Introduction

The I/O line configuration freeze function can reinforce the protection against the effects of an abnormal access resulting from a Single-event upset that may corrupt the value of one bit on the system bus during an access to the PIO or any other peripheral. Freezing the configuration of an I/O line prevents an unexpected access from modifying the configuration of the I/O line. Once the freeze is done, the I/O line configuration cannot be modified whatever software sequence is performed on the PIO.

36.5.16.2 Software Freeze

Once the I/O line configuration is done, it can be frozen by using the PIO I/O Freeze Configuration Register (PIO_IOFRx) of the corresponding group or the Secure PIO I/O Freeze Configuration Register (S_PIO_IOFRx) if the I/O line is Secure.

36.5.16.2.1 Physical Freeze

Setting PIO_IOFR.FPHY freezes the following fields (configured in PIO_CFGRx) of the Non-Secure I/O lines if the corresponding MSKx bit is set in PIO_MSKRx:

- FUNC: I/O Line Function
- DIR: Direction
- PUEN: Pull-Up Enable
- PDEN: Pull-Down Enable
- OPD: Open-Drain
- SCHMITT: Schmitt Trigger
- DRVSTR: Drive Strength

For Secure I/O lines, use the FPHY bit of the S_PIO_IOFRx and the S_PIO_MSKRx to freeze the fields above.

When the physical freeze is currently active on an I/O line, the PCFS flag is set when reading the PIO_CFGRx of the I/O line if the corresponding line is Non-Secure or the S_PIO_CFGRx if the I/O line is Secure.

Only a hardware reset can release fields listed above.

36.5.16.2.2 Interrupt Freeze

Setting PIO_IOFRx.FINT freezes the following fields (configured in PIO_CFGRx) of the Non-Secure I/O lines if the corresponding MSKx bit is set in PIO_MSKRx:

- IFEN: Input Filter Enable
- IFSCEN: Input Filter Slow Clock Enable
- EVTSEL: Event Selection

For Secure I/O lines, use S_PIO_IOFRx.FINT and the S_PIO_MSKRx to freeze the fields above.

When the “Interrupt Freeze” is currently active on an I/O line, the ICFS flag is set when reading the PIO_CFGRx of the I/O line (or the S_PIO_CFGRx if the I/O line is Secure).

Only a hardware reset can release fields listed above.

36.5.16.3 Tamper Freeze

For safety reasons, each I/O line can be configured to switch automatically in GPIO Input mode with pull-up in case of an external tamper event. This can be done by setting the TAMPEN bit in the PIO_CFGRx (or the S_PIO_CFGRx for a Secure I/O line). If the TAMPEN bit is set, the corresponding I/O line(s) switches automatically in General Purpose IO mode (GPIO) and Pull-Up is activated as long as the tamper event is detected. The “Tamper Freeze” is released when the tamper event is no longer present and frozen I/O line switch back to their applicative configuration automatically.

36.5.17 Register Write Protection

To prevent any single software error from corrupting PIO behavior, certain registers in the address space can be write-protected by setting WPEN and WPITEN in the PIO Write Protection Mode Register (PIO_WPMR) or the Secure PIO Write Protection Mode Register (S_PIO_WPMR).

If a write access to a Non-Secure write-protected register is detected, the WPVS flag in the PIO Write Protection Status Register (PIO_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

If a write access to a Secure write-protected register is detected, the WPVS flag in the Secure PIO Write Protection Status Register (S_PIO_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The respective WPVS bit is automatically cleared after reading the PIO_WPSR or S_PIO_WPSR.

The following registers are write-protected when WPEN is set in PIO_WPMR:

- PIO Mask Register
- PIO Configuration Register

The following registers are write-protected when WPEN is set in S_PIO_WPMR:

- Secure PIO Mask Register
- Secure PIO Configuration Register
- Secure PIO Slow Clock Divider Debouncing Register

The following registers are write-protected when WPITEN is set in PIO_WPMR:

- PIO Interrupt Enable Register
- PIO Interrupt Disable Register

The following registers are write-protected when WPITEN is set in S_PIO_WPMR:

- Secure PIO Interrupt Enable Register
- Secure PIO Interrupt Disable Register

36.6 I/O Lines Programming Example

The programming example shown in the table below is used to obtain the following configurations:

- PIOA Configuration:

- 4-bit output port on Secure I/O lines 0 to 3, open-drain, with pull-up resistor
- Four output signals on Non-Secure I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pull-up resistor, no pull-down resistor
- Secure I/O lines 16 to 19 assigned to peripheral A functions with pull-up resistor
- Non-Secure I/O lines 20 to 23 assigned to peripheral B functions with pull-down resistor

- PIOB Configuration:
  - Four input signals on Secure I/O lines 0 to 3 (to read push-button states for example), with pull-up resistors, glitch filters and input change interrupts
  - Four input signals on Non-Secure I/O lines 12 to 15 to read an external device status (polled, thus no input change interrupt), no pull-up resistor, no glitch filter
  - Secure I/O lines 16 to 23 assigned to peripheral B functions with pull-down resistor
  - Non-Secure I/O lines 24 to 27 assigned to peripheral D with Input Change Interrupt, no pull-up resistor and no pull-down resistor

Table 36-3. Programming Example

Action	Register	Value to be Written
PIOA: Set I/O lines 0 to 3 and 16 to 19 as Secure	S_PIO_SIOSR0 (offset 0x1034)	0x000F000F
PIOA: Set I/O lines 4 to 7 and 20 to 23 as Non-Secure	S_PIO_SIONR0 (offset 0x1030)	0x00F000F0
PIOA: 4-bit output port on Secure I/O lines 0 to 3, open-drain, with pull-up resistor	S_PIO_MSKR0 (offset 0x1000)	0x0000000F
	S_PIO_CFGR0 (offset 0x1004)	0x00004300
PIOA: Four output signals on Non-Secure I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pull-up resistor, no pull-down resistor	PIO_MSKR0 (offset 0x0)	0x000000F0
	PIO_CFGR0 (offset 0x4)	0x00000100
PIOA: Secure I/O lines 16 to 19 assigned to peripheral A functions with pull-up resistor	S_PIO_MSKR0 (offset 0x1000)	0x000F0000
	S_PIO_CFGR0 (offset 0x1004)	0x00000201
PIOA: Non-Secure I/O lines 20 to 23 assigned to peripheral B functions with pull-down resistor	PIO_MSKR0 (offset 0x0)	0x00F00000
	PIO_CFGR0 (offset 0x4)	0x00000402
PIOB: Set I/O lines 0 to 3 and 16 to 23as Secure	S_PIO_SIOSR1 (offset 0x1074)	0x00FF000F
PIOB: Set I/O lines 12 to 15 and 24 to 27 as Non-Secure	S_PIO_SIONR1 (offset 0x1070)	0x0F00F000
PIOB: Four input signals on Secure I/O lines 0 to 3 (to read push-button states for example), with pull-up resistors, glitch filters and interrupts on rising edge	S_PIO_MSKR1 (offset 0x1040)	0x0000000F
	S_PIO_CFGR1 (offset 0x1044)	0x01001200

...........continued

Action	Register	Value to be Written
PIOB: Four input signals on Non-Secure I/O line 12 to 15 to read an external device status (polled, thus no input change interrupt), no pull-up resistor, no glitch filter	PIO_MSKR1 (offset 0x40)	0x0000F000
	PIO_CFGR1 (offset 0x44)	0x01001200
PIOB: Secure I/O lines 16 to 23 assigned to peripheral B functions with pull-down resistor	S_PIO_MSKR1 (offset 0x1040)	0x00FF0000
	S_PIO_CFGR1 (offset 0x1044)	0x00000402
PIOB: Non-Secure I/O line 24 to 27 assigned to peripheral D with Input Interrupt on both edges, no pull-up resistor and no pull-down resistor	PIO_MSKR1 (offset 0x40)	0x0F000000
	PIO_CFGR1 (offset 0x44)	0x02000004
PIOB: Enable interrupt	S_PIO_IER1 (offset 0x1060)	0x0000000F
	PIO_IER1 (offset 0x60)	0x0F000000

36.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PIO_MSKR0	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x04	PIO_CFGR0	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x08	PIO_PDSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0C	PIO_LOCKSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10	PIO_SODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x14	PIO_CODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x18	PIO_ODSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1C ... 0x1F	Reserved									
0x20	PIO_IER0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x24	PIO_IDR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x28	PIO_IMR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x2C	PIO_ISR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x30 ... 0x3B	Reserved									
0x3C	PIO_IOFR0	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	PIO_MSKR1	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x44	PIO_CFGR1	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x48	PIO_PDSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x4C	PIO_LOCKSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x50	PIO_SODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x54	PIO_CODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x58	PIO_ODSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x5C ... 0x5F	Reserved									
0x60	PIO_IER1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x64	PIO_IDR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x68	PIO_IMR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x6C	PIO_ISR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x70 ... 0x7B	Reserved									
0x7C	PIO_IOFR1	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x80	PIO_MSKR2	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x84	PIO_CFGR2	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x88	PIO_PDSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x8C	PIO_LOCKSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x90	PIO_SODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x94	PIO_CODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x98	PIO_ODSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x9C ... 0x9F	Reserved									
0xA0	PIO_IER2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA4	PIO_IDR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA8	PIO_IMR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xAC	PIO_ISR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xB0 ... 0xBB	Reserved									
0xBC	PIO_IOFR2	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0xC0	PIO_MSKR3	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0xC4	PIO_CFGR3	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
		7:0						FUNC[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC8	PIO_PDSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xCC	PIO_LOCKSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xD0	PIO_SODR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xD4	PIO_CODR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xD8	PIO_ODSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xDC ... 0xDF	Reserved									
0xE0	PIO_IER3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xE4	PIO_IDR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xE8	PIO_IMR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xEC	PIO_ISR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xF0 ... 0xFB	Reserved									
0xFC	PIO_IOFR3	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x0100	PIO_MSKR4	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x0104	PIO_CFGR4	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x0108	PIO_PDSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010C	PIO_LOCKSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0110	PIO_SODR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0114	PIO_CODR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0118	PIO_ODSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x011C ... 0x011F	Reserved									
0x0120	PIO_IER4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0124	PIO_IDR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0128	PIO_IMR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x012C	PIO_ISR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0130 ... 0x013B	Reserved									
0x013C	PIO_IOFR4	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x0140 ... 0x05CF	Reserved									
0x05D0	PIO_TPFR	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0								TAMPFRZ
0x05D4 ... 0x05DF	Reserved									
0x05E0	PIO_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x05E4	PIO_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x05E8 ... 0x0FFF	Reserved									
0x1000	S_PIO_MSKR0	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1004	S_PIO_CFGR0	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1008	S_PIO_PDSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x100C	S_PIO_LOCKSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1010	S_PIO_SODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1014	S_PIO_CODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1018	S_PIO_ODSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x101C ... 0x101F	Reserved									
0x1020	S_PIO_IER0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1024	S_PIO_IDR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1028	S_PIO_IMR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x102C	S_PIO_ISR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1030	S_PIO_SIONR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x1034	S_PIO_SIOSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1038	S_PIO_IOSSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x103C	S_PIO_IOFR0	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1040	S_PIO_MSKR1	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1044	S_PIO_CFGR1	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1048	S_PIO_PDSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x104C	S_PIO_LOCKSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1050	S_PIO_SODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1054	S_PIO_CODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1058	S_PIO_ODSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x105C ... 0x105F	Reserved									
0x1060	S_PIO_IER1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1064	S_PIO_IDR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1068	S_PIO_IMR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x106C	S_PIO_ISR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x1070	S_PIO_SIONR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1074	S_PIO_SIOSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1078	S_PIO_IOSSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x107C	S_PIO_IOFR1	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1080	S_PIO_MSKR2	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1084	S_PIO_CFGR2	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1088	S_PIO_PDSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x108C	S_PIO_LOCKSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1090	S_PIO_SODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1094	S_PIO_CODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1098	S_PIO_ODSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x109C ... 0x109F	Reserved									
0x10A0	S_PIO_IER2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10A4	S_PIO_IDR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10A8	S_PIO_IMR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10AC	S_PIO_ISR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10B0	S_PIO_SIONR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10B4	S_PIO_SIOSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10B8	S_PIO_IOSSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10BC	S_PIO_IOFR2	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x10C0	S_PIO_MSKR3	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x10C4	S_PIO_CFGR3	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x10C8	S_PIO_PDSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10CC	S_PIO_LOCKSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10D0	S_PIO_SODR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10D4	S_PIO_CODR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10D8	S_PIO_ODSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10DC ... 0x10DF	Reserved									
0x10E0	S_PIO_IER3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10E4	S_PIO_IDR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10E8	S_PIO_IMR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10EC	S_PIO_ISR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10F0	S_PIO_SIONR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10F4	S_PIO_SIOSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10F8	S_PIO_IOSSR3	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10FC	S_PIO_IOFR3	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1100	S_PIO_MSKR4	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1104	S_PIO_CFGR4	31:24	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
		23:16							DRVSTR[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1108	S_PIO_PDSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x110C	S_PIO_LOCKSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1110	S_PIO_SODR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1114	S_PIO_CODR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1118	S_PIO_ODSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x111C ... 0x111F	Reserved									
0x1120	S_PIO_IER4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x1124	S_PIO_IDR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1128	S_PIO_IMR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x112C	S_PIO_ISR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1130	S_PIO_SIONR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1134	S_PIO_SIOSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1138	S_PIO_IOSSR4	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x113C	S_PIO_IOFR4	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1140 ... 0x14FF	Reserved									
0x1500	S_PIO_SCDR	31:24								
		23:16								
		15:8			DIV[13:8]					
		7:0	DIV[7:0]							
0x1504 ... 0x15CF	Reserved									
0x15D0	S_PIO_TPFR	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0								TAMPFRZ
0x15D4 ... 0x15DF	Reserved									
0x15E0	S_PIO_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x15E4	S_PIO_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

36.7.1 PIO Mask Register

Name: PIO_MSKRx
Offset: 0x00 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – MSKy PIO Line y Mask
These bits define the I/O lines to be configured when writing the PIO Configuration Register.
0 (DISABLED): Writing the PIO_CFGRx, PIO_ODSRx or PIO_IOFRx does not affect the corresponding I/O line configuration.
1 (ENABLED): Writing the PIO_CFGRx, PIO_ODSRx or PIO_IOFRx updates the corresponding I/O line configuration.

36.7.2 PIO Configuration Register

Name: PIO_CFGRx
Offset: 0x04 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
Access	R/W	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
							DRVSTR[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCHMITT	OPD	IFSCEN	IFEN	SR	PDEN	PUEN	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						FUNC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 31 – TAMPEN Tamper Enable
Defines the action to do in case of tamper event for the I/O lines of the I/O group x defined in the PIO Mask Register.
0 (NO_FREEZE): No effect on the selected I/O lines.
1 (FREEZE): The selected I/O lines are automatically set in GPIO input with pull-up in case of a tamper event detection.

Bit 30 – ICFS Interrupt Configuration Freeze Status (read-only)
Gives information about the freeze state of the following fields of the read I/O line configuration:
• IFEN: Input Filter Enable
• IFSCEN: Input Filter Slow Clock Enable
• EVTSEL: Event Selection
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bit 29 – PCFS Physical Configuration Freeze Status (read-only)
Gives information about the freeze state of the following fields of the read I/O line configuration:
• FUNC: I/O Line Function
• DIR: Direction
• PUEN: Pull-Up Enable
• PDEN: Pull-Down Enable
• OPD: Open-Drain
• SCHMITT: Schmitt Trigger
• DRVSTR: Drive Strength
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bits 26:24 – EVTSEL[2:0] Event Selection
Defines the type of event to detect on the I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	FALLING	Event detection on input falling edge
1	RISING	Event detection on input rising edge
2	BOTH	Event detection on input both edge
3	LOW	Event detection on low level input
4	HIGH	Event detection on high level input
5	–	Reserved
6	–	Reserved
7	–	Reserved

Bits 17:16 – DRVSTR[1:0] Drive Strength
Defines the drive strength of the I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	LOW_OR_TYPD	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else Type D drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
1	HIGH_OR_TYPA	High drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else Type A drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
2	LOW_OR_TYPC	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else Type C drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
3	LOW_OR_TYPB	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else Type B drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.

Bit 15 – SCHMITT Schmitt Trigger
Defines the Schmitt trigger configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (ENABLED): Schmitt trigger is enabled for the selected I/O lines.
1 (DISABLED): Schmitt trigger is disabled for the selected I/O lines.

Bit 14 – OPD Open Drain
Defines the open drain configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The open-drain is disabled for the selected I/O lines. I/O lines are driven at high- and low-level.
1 (ENABLED): The open-drain is enabled for the selected I/O lines. I/O lines are driven at low-level only.

Bit 13 – IFSCEN Input Filter Slow Clock Enable
Defines the clock source of the glitch filtering for the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The glitch filter is able to filter glitches with a duration less than 1 peripheral clock cycle for the selected I/O lines.
1 (ENABLED): The debouncing filter is able to filter pulses with a duration less than 1 divided slow clock cycle for the selected I/O lines.

Bit 12 – IFEN Input Filter Enable
Defines if the glitch filtering is used for the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The input filter is disabled for the selected I/O lines.
1 (ENABLED): The input filter is enabled for the selected I/O lines.

Bit 11 – SR Slew Rate
Slew rate control does not apply to high-speed I/Os.
0 (DISABLED): Slew rate control is disabled for the selected I/O lines.
1 (ENABLED): Slew rate control is enabled for the selected I/O lines.

Bit 10 – PDEN Pull-Down Enable
Defines the pull-down configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
PDEN can be written to 1 only if PUEN is written to 0.
0 (DISABLED): Pull-down is disabled for the selected I/O lines.
1 (ENABLED): Pull-down is enabled for the selected I/O lines only if PUEN is 0.

Bit 9 – PUEN Pull-Up Enable
Defines the pull-up configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): Pull-up is disabled for the selected I/O lines.
1 (ENABLED): Pull-up is enabled for the selected I/O lines.

Bit 8 – DIR Direction
Defines the direction of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (INPUT): The selected I/O lines are pure inputs.
1 (OUTPUT): The selected I/O lines are enabled in output.

Bits 2:0 – FUNC[2:0] I/O Line Function
Defines the function for I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	GPIO	Selects the PIO mode for the selected I/O lines.
1	PERIPH_A	Selects peripheral A for the selected I/O lines.
2	PERIPH_B	Selects peripheral B for the selected I/O lines.
3	PERIPH_C	Selects peripheral C for the selected I/O lines.
4	PERIPH_D	Selects peripheral D for the selected I/O lines.
5	PERIPH_E	Selects peripheral E for the selected I/O lines.
6	PERIPH_F	Selects peripheral F for the selected I/O lines.
7	PERIPH_G	Selects peripheral G for the selected I/O lines.

36.7.3 PIO Pin Data Status Register

Name: PIO_PDSRx
Offset: 0x08 + x*0x40 [x=0..4]
Reset: –
Property: Read-only

Reset value of PIO_PDSR depends on the level of the I/O lines. Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Data Status

Value	Description
0	The I/O line of the I/O group x is at level 0.
1	The I/O line of the I/O group x is at level 1.

36.7.4 PIO Lock Status Register

Name: PIO_LOCKSRx
Offset: 0x0C + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Lock Status

Value	Description
0	The I/O line of the I/O group x is not locked.
1	The I/O line of the I/O group x is locked.

36.7.5 PIO Set Output Data Register

Name: PIO_SODRx
Offset: 0x10 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line of I/O group x.

36.7.6 PIO Clear Output Data Register

Name: PIO_CODRx
Offset: 0x14 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line of the I/O group x.

36.7.7 PIO Output Data Status Register

Name: PIO_ODSRx
Offset: 0x18 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Output Data Status

Value	Description
0	The data to be driven on the I/O line of the I/O group x is 0.
1	The data to be driven on the I/O line of the I/O group x is 1.

36.7.8 PIO Interrupt Enable Register

Name: PIO_IERx
Offset: 0x20 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the Input Change interrupt on the I/O line of the I/O group x.

36.7.9 PIO Interrupt Disable Register

Name: PIO_IDRx
Offset: 0x24 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the Input Change interrupt on the I/O line of the I/O group x.

36.7.10 PIO Interrupt Mask Register

Name: PIO_IMRx
Offset: 0x28 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Mask

Value	Description
0	Input Change interrupt is disabled on the I/O line of the I/O group x.
1	Input Change interrupt is enabled on the I/O line of the I/O group x.

36.7.11 PIO Interrupt Status Register

Name: PIO_ISRx
Offset: 0x2C + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

PIO_ISR is reset at 0x00000000. However, the first read of the register may read a different value as input changes may have occurred.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Status

Value	Description
0	No Input Change has been detected on the I/O line of the I/O group x since PIO_ISRx was last read or since reset.
1	At least one Input Change has been detected on the I/O line of the I/O group since PIO_ISRx was last read or since reset.

36.7.12 PIO I/O Freeze Configuration Register

Name: PIO_IOFRx
Offset: 0x3C + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
							FINT	FPHY
Access							W	W
Reset							–	–

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
0x494F46	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit.

Bit 1 – FINT Freeze Interrupt Configuration
Only a hardware reset can reset the FINT bit.

Value	Description
0	No effect.
1	Freezes the following configuration fields of Non-Secure I/O lines if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • IFEN: Input Filter Enable • IFSCEN: Input Filter Slow Clock Enable • EVTSEL: Event Selection

Bit 0 – FPHY Freeze Physical Configuration
Only a hardware reset can reset the FPHY bit.

Value	Description
0	No effect.

Value	Description
1	Freezes the following configuration fields of Non-Secure I/O lines if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • FUNC: I/O Line Function • DIR: Direction • PUEN: Pull-Up Enable • PDEN: Pull-Down Enable • OPD: Open-Drain • SCHMITT: Schmitt Trigger • DRVSTR: Drive Strength

36.7.13 PIO Tamper Freeze Register

Name: PIO_TPFR
Offset: 0x5D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								TAMPFRZ
Access								R/W
Reset								0

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
0x545046	PASSWD	Writing any other value in this field aborts the write operation of the TAMPFRZ bit.

Bit 0 – TAMPFRZ Freeze on Tamper Event
On tamper detection, Non-Secure I/O lines with TAMPEN bit at 1 will be forced as GPIO inputs with pull-up.
Only a hardware reset can reset the TAMPFRZ bit.

Value	Description
0	No effect.
1	Enables freeze of Non-Secure I/O lines on tamper event detection if FRZKEY corresponds to 0x545046 (“TPF” in ASCII).

36.7.14 PIO Write Protection Mode Register

Name: PIO_WPMR
Offset: 0x5E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

36.7.15 PIO Write Protection Status Register

Name: PIO_WPSR
Offset: 0x5E4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PIO_WPSR.
1	A write protection violation has occurred since the last read of the PIO_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

36.7.16 Secure PIO Mask Register

Name: S_PIO_MSKRx
Offset: 0x1000 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Secure PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – MSKy PIO Line y Mask
These bits define the I/O lines to be configured when writing the Secure PIO Configuration Register.
0 (DISABLED): Writing the S_PIO_CFGRx, S_PIO_ODSRx or S_PIO_IOFRx does not affect the corresponding I/O line configuration.
1 (ENABLED): Writing the S_PIO_CFGRx, S_PIO_ODSRx or S_PIO_IOFRX updates the corresponding I/O line configuration.

36.7.17 Secure PIO Configuration Register

Name: S_PIO_CFGRx
Offset: 0x1004 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Secure PIO Write Protection Mode Register.

Writing this register will only affect I/O lines enabled in the S_PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	TAMPEN	ICFS	PCFS			EVTSEL[2:0]		
Access	R/W	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
							DRVSTR[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
						FUNC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 31 – TAMPEN Tamper Enable
Defines the action to do in case of tamper event for the I/O lines of the I/O group x defined in the Secure PIO Mask Register.
0 (NO_FREEZE): No effect on the selected I/O lines.
1 (FREEZE): The selected I/O lines are automatically set in GPIO input with pull-up in case of a tamper event detection.

Bit 30 – ICFS Interrupt Configuration Freeze Status
Gives information about the freeze state of the following fields of the read I/O line configuration:
• IFEN: Input Filter Enable
• IFSCEN: Input Filter Slow Clock Enable
• EVTSEL: Event Selection
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bit 29 – PCFS Physical Configuration Freeze Status
Gives information about the freeze state of the following fields of the read I/O line configuration:
• FUNC: I/O Line Function
• DIR: Direction
• PUEN: Pull-Up Enable
• PDEN: Pull-Down Enable
• OPD: Open-Drain
• SCHMITT: Schmitt Trigger
• DRVSTR: Drive Strength
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.

1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bits 26:24 – EVTSEL[2:0] Event Selection
Defines the type of event to detect on the I/O lines of the I/O group x according to the Secure PIO Mask Register.

Value	Name	Description
0	FALLING	Event detection on input falling edge
1	RISING	Event detection on input rising edge
2	BOTH	Event detection on input both edge
3	LOW	Event detection on low level input
4	HIGH	Event detection on high level input
5	–	Reserved
6	–	Reserved
7	–	Reserved

Bits 17:16 – DRVSTR[1:0] Drive Strength
Defines the drive strength of the I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	LOW_OR_HS100	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else lowest drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
1	HIGH_OR_HS33	High drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else highest drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
2	LOW_OR_HS66	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else middle-low drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.
3	LOW_OR_HS50	Low drive strength when the IO is driven in GPIO mode or by any non high-speed peripheral, else middle-high drive for high-speed peripherals. Refer to the section “Electrical Characteristics” for values.

Bit 15 – SCHMITT Schmitt Trigger
Defines the Schmitt trigger configuration of the I/O lines of the I/O group x according to the Secure PIO Mask Register.
0 (ENABLED): Schmitt trigger is enabled for the selected I/O lines.
1 (DISABLED): Schmitt trigger is disabled for the selected I/O lines.

Bit 14 – OPD Open Drain
Defines the open drain configuration of the I/O lines of the I/O group x according to the Secure PIO Mask Register.
0 (DISABLED): The open drain is disabled for the selected I/O lines. I/O lines are driven at high- and low-level.
1 (ENABLED): The open drain is enabled for the selected I/O lines. I/O lines are driven at low-level only.

Bit 13 – IFSCEN Input Filter Slow Clock Enable
Defines the clock source of the glitch filtering for the I/O lines of the I/O group x according to the Secure PIO Mask Register.

Value	Description
0	The glitch filter is able to filter glitches with a duration less than 1 peripheral clock cycle for the selected I/O lines.
1	The debouncing filter is able to filter pulses with a duration less than 1 divided slow clock cycle for the selected I/O lines.

Bit 12 – IFEN Input Filter Enable
Defines if the glitch filtering is used for the I/O lines of the I/O group x according to the Secure PIO Mask Register.
0 (DISABLED): The input filter is disabled for the selected I/O lines.
1 (ENABLED): The input filter is enabled for the selected I/O lines.

Bit 10 – PDEN Pull-Down Enable
Defines the pull-down configuration of the I/O lines of the I/O group x according to the Secure PIO Mask Register.
PDEN can be written to 1 only if PUEN is written to 0.
0 (DISABLED): Pull-down is disabled for the selected I/O lines.
1 (ENABLED): Pull-down is enabled for the selected I/O lines only if PUEN is 0.

Bit 9 – PUEN Pull-Up Enable
Defines the pull-up configuration of the I/O lines of the I/O group x according to the Secure PIO Mask Register.
0 (DISABLED): Pull-up is disabled for the selected I/O lines.
1 (ENABLED): Pull-up is enabled for the selected I/O lines.

Bit 8 – DIR Direction
Defines the direction of the I/O lines of the I/O group x according to the Secure PIO Mask Register.
0 (INPUT): The selected I/O lines are pure inputs.
1 (OUTPUT): The selected I/O lines are enabled in output.

Bits 2:0 – FUNC[2:0] I/O Line Function
Defines the function for I/O lines of the I/O group x according to the Secure PIO Mask Register.

Value	Name	Description
0	GPIO	Selects the PIO mode for the selected I/O lines.
1	PERIPH_A	Selects peripheral A for the selected I/O lines.
2	PERIPH_B	Selects peripheral B for the selected I/O lines.
3	PERIPH_C	Selects peripheral C for the selected I/O lines.
4	PERIPH_D	Selects peripheral D for the selected I/O lines.
5	PERIPH_E	Selects peripheral E for the selected I/O lines.
6	PERIPH_F	Selects peripheral F for the selected I/O lines.
7	PERIPH_G	Selects peripheral G for the selected I/O lines.

36.7.18 Secure PIO Pin Data Status Register

Name: S_PIO_PDSRx
Offset: 0x1008 + x*0x40 [x=0..4]
Reset: –
Property: Read-only

Reset value of PIO_PDSR and S_PIO_PDSR depends on the level of the I/O lines. Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Data Status

Value	Description
0	The I/O line of the I/O group x is at level 0.
1	The I/O line of the I/O group x is at level 1.

36.7.19 Secure PIO Lock Status Register

Name: S_PIO_LOCKSRx
Offset: 0x100C + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Lock Status

Value	Description
0	The I/O line of the I/O group x is not locked.
1	The I/O line of the I/O group x is locked.

36.7.20 Secure PIO Set Output Data Register

Name: S_PIO_SODRx
Offset: 0x1010 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line of I/O group x.

36.7.21 Secure PIO Clear Output Data Register

Name: S_PIO_CODRx
Offset: 0x1014 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line of the I/O group x.

36.7.22 Secure PIO Output Data Status Register

Name: S_PIO_ODSRx
Offset: 0x1018 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read/Write

Writing this register will only affect I/O lines enabled in the S_PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Output Data Status

Value	Description
0	The data to be driven on the I/O line of the I/O group x is 0.
1	The data to be driven on the I/O line of the I/O group x is 1.

36.7.23 Secure PIO Interrupt Enable Register

Name: S_PIO_IERx
Offset: 0x1020 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Secure PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the Input Change interrupt on the I/O line of the I/O group x.

36.7.24 Secure PIO Interrupt Disable Register

Name: S_PIO_IDRx
Offset: 0x1024 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Secure PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the Input Change interrupt on the I/O line of the I/O group x.

36.7.25 Secure PIO Interrupt Mask Register

Name: S_PIO_IMRx
Offset: 0x1028 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Mask

Value	Description
0	Input Change interrupt is disabled on the I/O line of the I/O group x.
1	Input Change interrupt is enabled on the I/O line of the I/O group x.

36.7.26 Secure PIO Interrupt Status Register

Name: S_PIO_ISRx
Offset: 0x102C + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

PIO_ISR and S_PIO_ISR are reset at 0x000000000. However, the first read of the register may read a different value as input changes may have occurred.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Input Change Interrupt Status

Value	Description
0	No Input Change has been detected on the I/O line of the I/O group x since S_PIO_ISRx was last read or since reset.
1	At least one Input Change has been detected on the I/O line of the I/O group since S_PIO_ISRx was last read or since reset.

36.7.27 Secure PIO Set I/O Non-Secure Register

Name: S_PIO_SIONRx
Offset: 0x1030 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Set I/O Non-Secure

Value	Description
0	No effect.
1	Set the I/O line of the I/O group x in Non-Secure mode.

36.7.28 Secure PIO Set I/O Secure Register

Name: S_PIO_SIOSRx
Offset: 0x1034 + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
Set I/O Secure

Value	Description
0	No effect.
1	Set the I/O line of the I/O group x in Secure mode.

36.7.29 Secure PIO I/O Security Status Register

Name: S_PIO_IOSSRx
Offset: 0x1038 + x*0x40 [x=0..4]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px
I/O Security Status
0 (SECURE): The I/O line of the I/O group x is in Secure mode.
1 (NON_SECURE): The I/O line of the I/O group x is in Non-Secure mode.

36.7.30 Secure PIO I/O Freeze Configuration Register

Name: S_PIO_IOFRx
Offset: 0x103C + x*0x40 [x=0..4]
Reset: –
Property: Write-only

Writing this register will only affect I/O lines enabled in the S_PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
							FINT	FPHY
Access							W	W
Reset							–	–

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
0x494F46	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit.

Bit 1 – FINT Freeze Interrupt Configuration
Only a hardware reset can reset the FINT bit.

Value	Description
0	No effect.
1	Freezes the following configuration fields of Secure I/O lines if FRZKEY corresponds to 0x494F46 ("IOF" in ASCII): • IFEN: Input Filter Enable • IFSCEN: Input Filter Slow Clock Enable • EVTSEL: Event Selection

Bit 0 – FPHY Freeze Physical Configuration
Only a hardware reset can reset the FPHY bit.

Value	Description
0	No effect.

Value	Description
1	Freezes the following configuration fields of Secure I/O lines if FRZKEY corresponds to 0x494F46 ("IOF" in ASCII): • FUNC: I/O Line Function • DIR: Direction • PUEN: Pull-Up Enable • PDEN: Pull-Down Enable • OPD: Open-Drain • SCHMITT: Schmitt Trigger • DRVSTR: Drive Strength

36.7.31 Secure PIO Slow Clock Divider Debouncing Register

Name: S_PIO_SCDR
Offset: 0x1500
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Secure PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			DIV[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – DIV[13:0] Slow Clock Divider Selection for Debouncing
$t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$

36.7.32 Secure PIO Tamper Freeze Register

Name: S_PIO_TPFR
Offset: 0x15D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								TAMPFRZ
Access								R/W
Reset								0

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
`0x545046`	PASSWD	Writing any other value in this field aborts the write operation of the TAMPFRZ bit.

Bit 0 – TAMPFRZ Freeze on Tamper Event

On tamper detection, Secure I/O lines with TAMPEN bit at 1 will be forced as GPIO inputs with pull-up.
Only a hardware reset can reset the TAMPFRZ bit.

Value	Description
`0`	No effect.
`1`	Enables to freeze Secure I/O lines on tamper event detection if FRZKEY corresponds to 0x545046 (“TPF” in ASCII).

36.7.33 Secure PIO Write Protection Mode Register

Name: S_PIO_WPMR
Offset: 0x15E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on secure interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection on secure interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

36.7.34 Secure PIO Write Protection Status Register

Name: S_PIO_WPSR
Offset: 0x15E4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

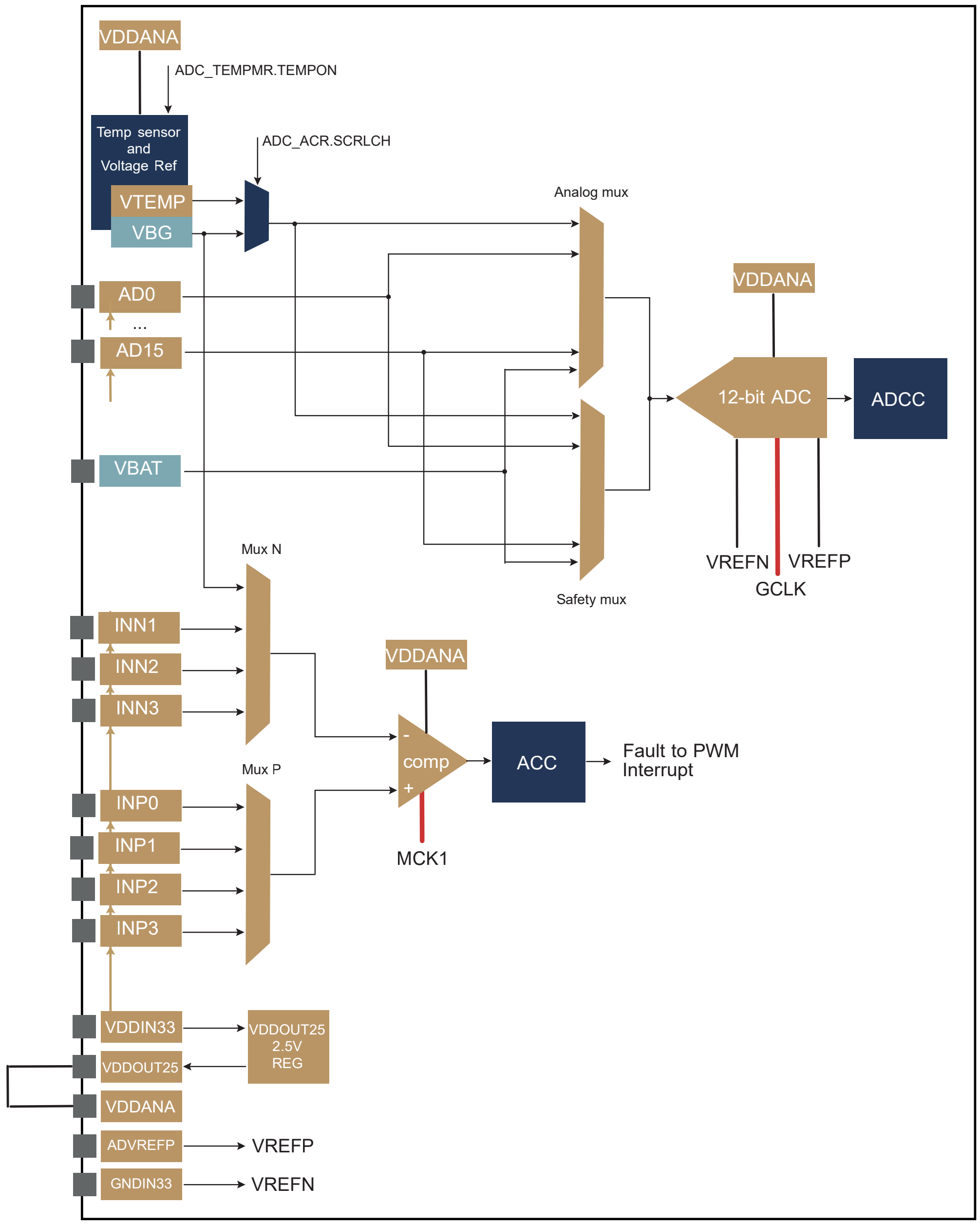
Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the S_PIO_WPSR.
1	A write protection violation has occurred since the last read of the S_PIO_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

37. ANALOG SUBSYSTEM

37.1 Block Diagram

Figure 37-1. Analog Subsystem Block Diagram



Before using VTEMP or VBG as ADC or ACC input, enable the temperature sensor and voltage reference with ADC_TEMPMR.TEMPON =1.

37.2 Components

- Analog-to-Digital Converter (ADC)

- Temperature sensor and reference block
- Analog Comparator (ACC)

37.3 Product Dependencies

37.3.1 Clocks

All clocks are controlled by the PMC, which is a part of the system controller.

ADC is only clocked by its GCLK. It must be started before any operation.

ACC is located on the APB Client (APS) matrix, clocked by MCK1.

37.3.2 Interrupts

Refer to the table Peripheral Identifiers.

37.3.3 Reset

Analog peripherals are connected to the processor and peripherals reset line.

37.3.4 I/Os

Analog I/Os are seven ACC inputs and 16 ADC inputs powered by VDDIN33/GNDIN33. To avoid any unwanted input loading, disable the pull-up and pull-down resistors before using these lines in the PIO_CFGRx and S_PIO_CFGRx registers.

37.4 Special Functions in SFR/SFRBU

None.

38. Analog-to-Digital Converter (ADC) Controller

38.1 Description

The ADC is based on a 12-bit Analog-to-Digital Converter (ADC) managed by an ADC Controller providing enhanced resolution up to 16 bits. The conversions extend from the voltage carried on pin GNDIN33 to the voltage carried on pin ADVREFP.

Conversion results are reported in a common register for all channels, as well as in a channel-dedicated register.

The 13-bit, 14-bit, 15-bit and 16-bit resolution modes are obtained by averaging multiple samples to decrease quantization noise. For the 13-bit mode, 4 samples are used, which gives a real sample rate of 1/4 of the actual sample frequency. For the 14-bit mode, 16 samples are used, giving a real sample rate of 1/16 of the actual sample frequency. For the 15-bit and 16-bit modes, respectively 64 and 256 samples are used, giving a real sample rate of respectively 1/64 and 1/256 of the actual sample frequency. This arrangement allows conversion speed to be traded off against for better accuracy.

The last channel is internally connected to the temperature sensor output voltage (VTEMP) or to the reference voltage (VBG). The selection is controlled in the analog control register (ADC_ACR).

The software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) can be selected for triggering a conversion.

The comparison circuitry allows automatic detection of values below a threshold, higher than a threshold, in a given range or outside the range, thresholds and ranges being fully configurable.

The ADC Controller internal fault output is directly connected to the PWM fault input. This input can be asserted by means of a comparison circuitry to immediately put the PWM output in a safe state (pure combinational path).

The ADC also integrates a Sleep mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

This ADC has a selectable single-ended or fully differential input.

Note: In this section, REFP = ADVREFP and REFN = GNDIN33.

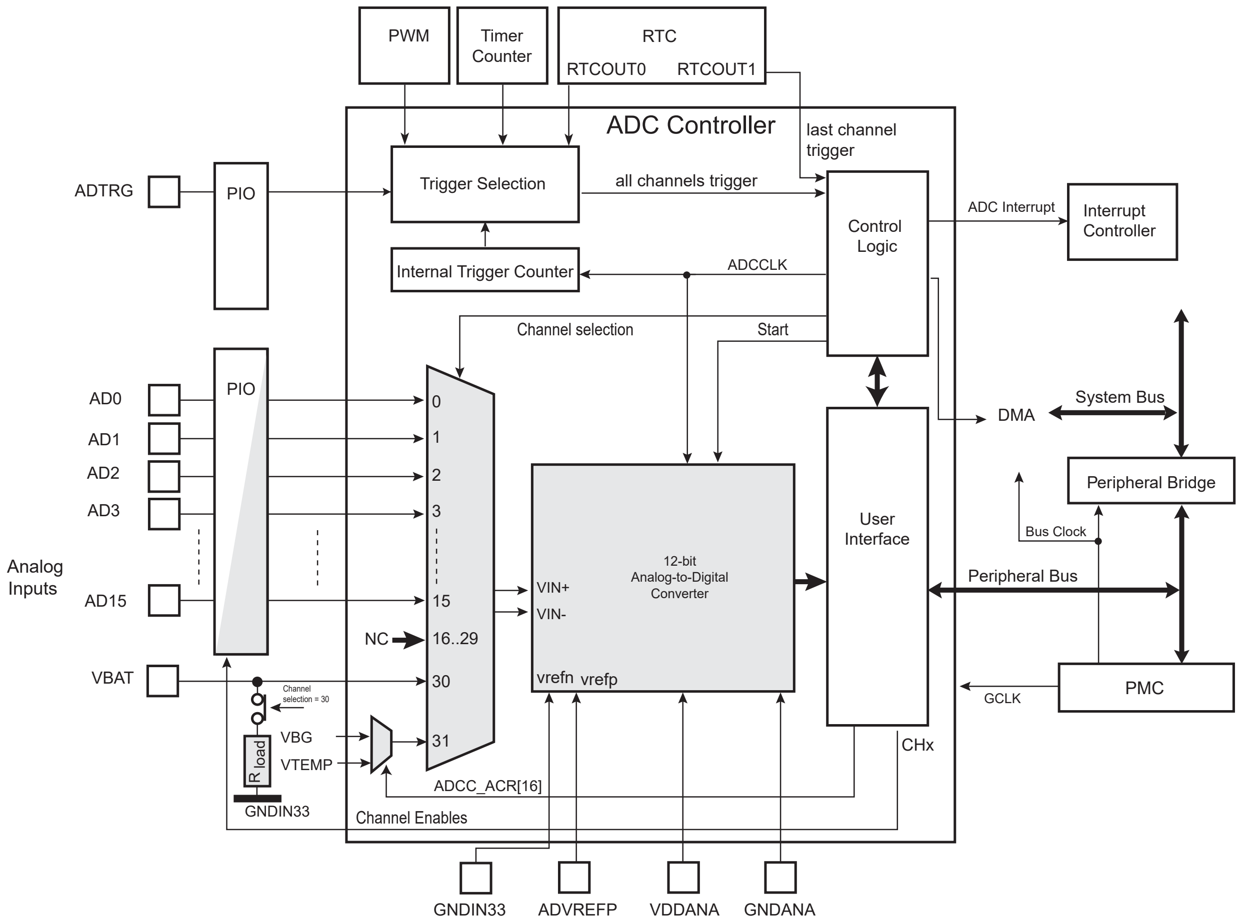
38.2 Embedded Characteristics

- 12-bit Resolution with Enhanced Mode up to 16 bits
- 1 MSps Conversion Rate
- Digital Averaging Function providing Enhanced Resolution Mode up to 16 bits
- On-chip Temperature Sensor Management
- Selectable Single-Ended or Differential Input Voltage
- Digital Correction of Offset and Gain Errors
- Individual Enable and Disable of Each Channel
- Hardware or Software Trigger from:
  - External trigger pin
  - ADC internal trigger counter
  - Timer Counter outputs
  - PWM event line
- Drive of PWM Fault Input
- DMA Support
- Two Sleep Modes (Automatic Wake-up on Trigger)
  - Lowest power consumption (voltage reference OFF between conversions)
  - Fast wake-up time response on trigger event (voltage reference ON between conversions)
- Channel Sequence Customizing

- Automatic Window Comparison of Converted Values
- Register Write Protection

38.3 Block Diagram

Figure 38-1. ADC Block Diagram



38.4 Product Dependencies

38.4.1 Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

38.4.2 Interrupt Sources

The ADC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the ADC interrupt requires the interrupt controller to be programmed first.

38.4.3 Battery Voltage

The battery voltage is internally connected to the channel with the highest index minus one. A fixed division factor is applied on VBAT voltage to fit the CH30 maximum input range (refer to the section “Electrical Characteristics” for the ratio value).

38.4.4 Temperature Sensor

The temperature sensor is internally connected to the channel with the highest index of the ADC.

The temperature sensor provides two output voltages VTEMP and VBG. VTEMP is proportional to the absolute temperature (PTAT) voltage and VBG is a quasi-temperature independent voltage.

To activate the temperature sensor, the bit TEMPON in the Temperature Sensor Mode register (ADC_TEMPMR) must be set. After setting the bit, the start-up time of the temperature sensor must be achieved prior to initiating any measurements.

38.4.5 I/O Lines

The digital inputs ADx and ADTRG are multiplexed with digital functions on the I/O lines. The analog mode for ADx I/O pins is automatically selected when the corresponding conversion channel is enabled. ADTRG is enabled by configuring the PIO controller.

38.4.6 Hardware Triggers

The ADC can use internal signals to start conversions. See the field TRGSEL for details on the wiring of internal triggers.

38.4.7 Fault Output

The ADC Controller has the FAULT output connected to the FAULT input of PWM. Refer to sections Fault Event and “Pulse Width Modulation Controller (PWM)”.

38.5 Functional Description

38.5.1 Analog-to-Digital Conversion

Once the programmed start-up time (ADC_MR.STARTUP) has elapsed, ADC conversions are sequenced by three operating times:

- Tracking time—the time for the ADC to charge its input sampling capacitor to the input voltage. When several channels are converted consecutively, the inherent tracking time is 6 ADC clock cycles. However, the tracking time can be increased using ADC_MR.TRACKTIM and the TRACKX field in the Extended Mode register (ADC_EMR).
- ADC inherent conversion time—the time for the ADC to convert the sampled analog voltage. This time is constant and is defined from start of conversion to end of conversion.
- Channel conversion period—the effective time between the end of the current channel conversion and the end of the next channel conversion.

Figure 38-2. Sequence of Consecutive ADC Conversions with TRACKTIM = 0

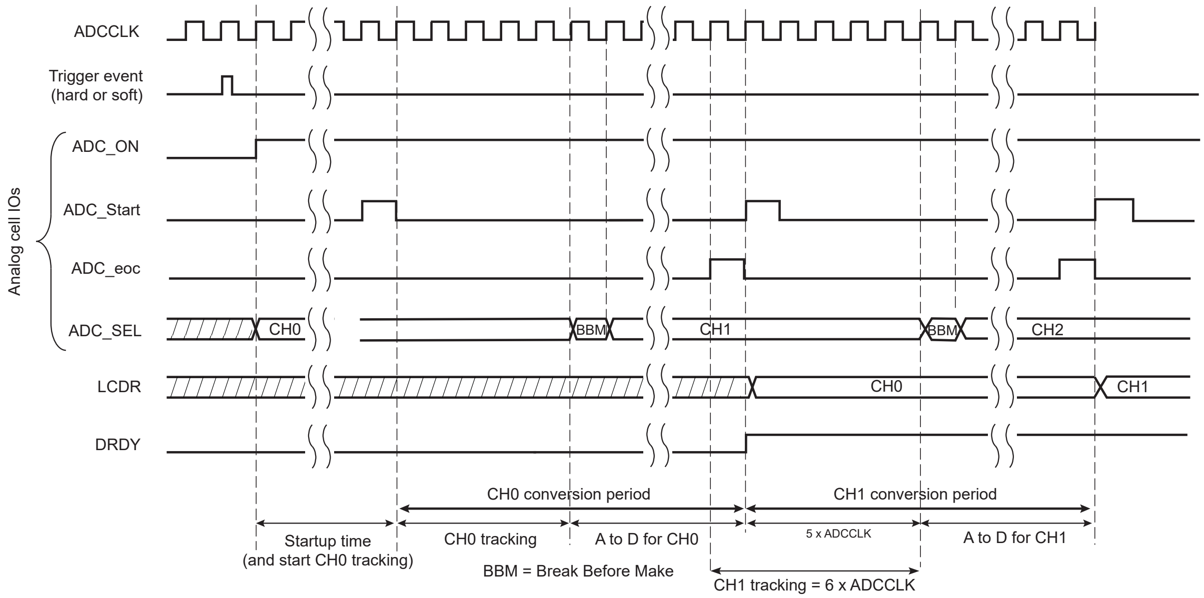
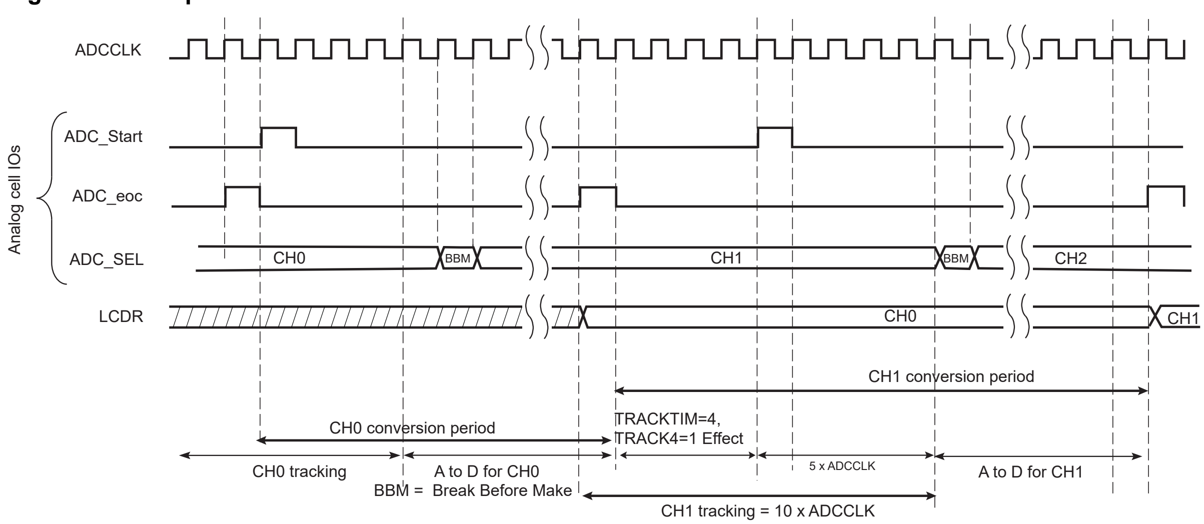


Figure 38-3. Sequence of Consecutive ADC Conversions with TRACKTIM = 4 and TRACKX = 1



38.5.2 ADC Clock

The ADC uses the ADC clock (ADCCLK) to perform conversions. The ADC clock frequency is selected in ADC_MR.PRESCAL.

To generate the ADC clock, the prescaler uses the peripheral clock driven by a GCLK clock.

ADC_MR.PRESCAL must be programmed to provide the ADC clock frequency parameter given in the section “Electrical Characteristics”.

38.5.3 ADC Reference Voltage

The voltage reference input of the ADC is the ADVREFP pin. Refer to the section “Electrical Characteristics” for further details.

38.5.4 Conversion Resolution

The ADC has a native resolution of 12 bits.

The ADC Controller provides enhanced resolution up to 16 bits by means of digital averaging.

If the digital input ADTRG is asynchronous to the ADC peripheral clock, the internal resynchronization introduces a jitter of 1 peripheral clock. This jitter may reduce the resolution of the converted signal.

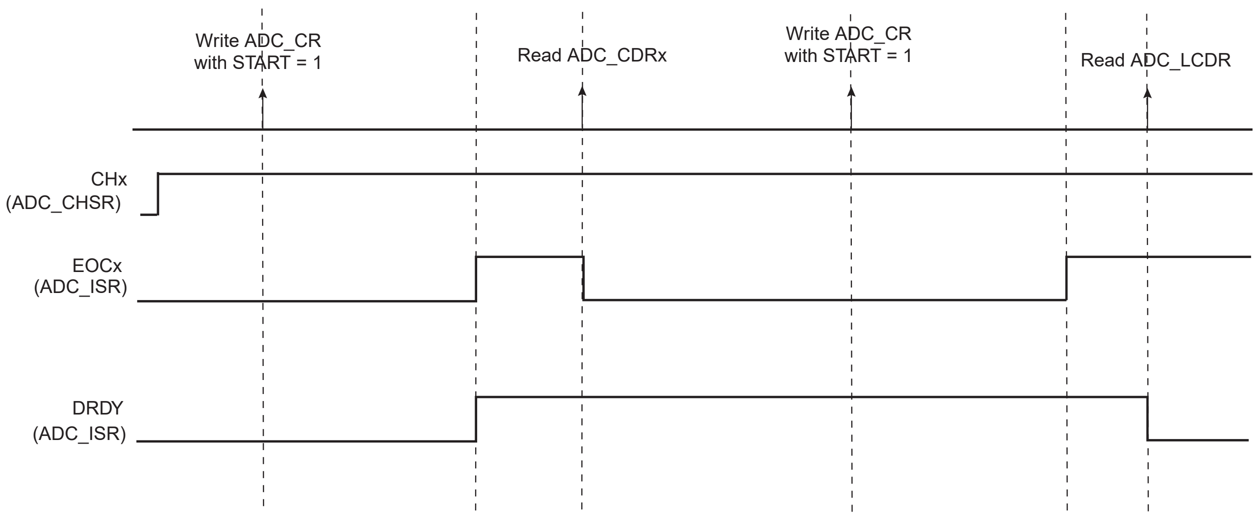
38.5.5 Conversion Results

When a conversion is completed, the resulting digital value is stored in the Channel Data register (ADC_CDRx) of the current channel and in the Last Converted Data register (ADC_LCDR). By setting ADC_EMR.TAG, ADC_LCDR presents the channel number associated with the last converted data in NO_OSR_CHNB/CHNBOSR.

When a conversion is completed, EOCx in the End of Conversion Interrupt Status register (ADC_EOC_ISR) and the bit DRDY in the Interrupt Status register (ADC_ISR) are set. In the case of a connected DMA channel, DRDY rising triggers a data request. In any case, either EOC or DRDY can trigger an interrupt.

Reading one of the ADC_CDRx clears the corresponding EOC. Reading ADC_LCDR clears DRDY.

Figure 38-4. EOCx and DRDY Flag Behavior

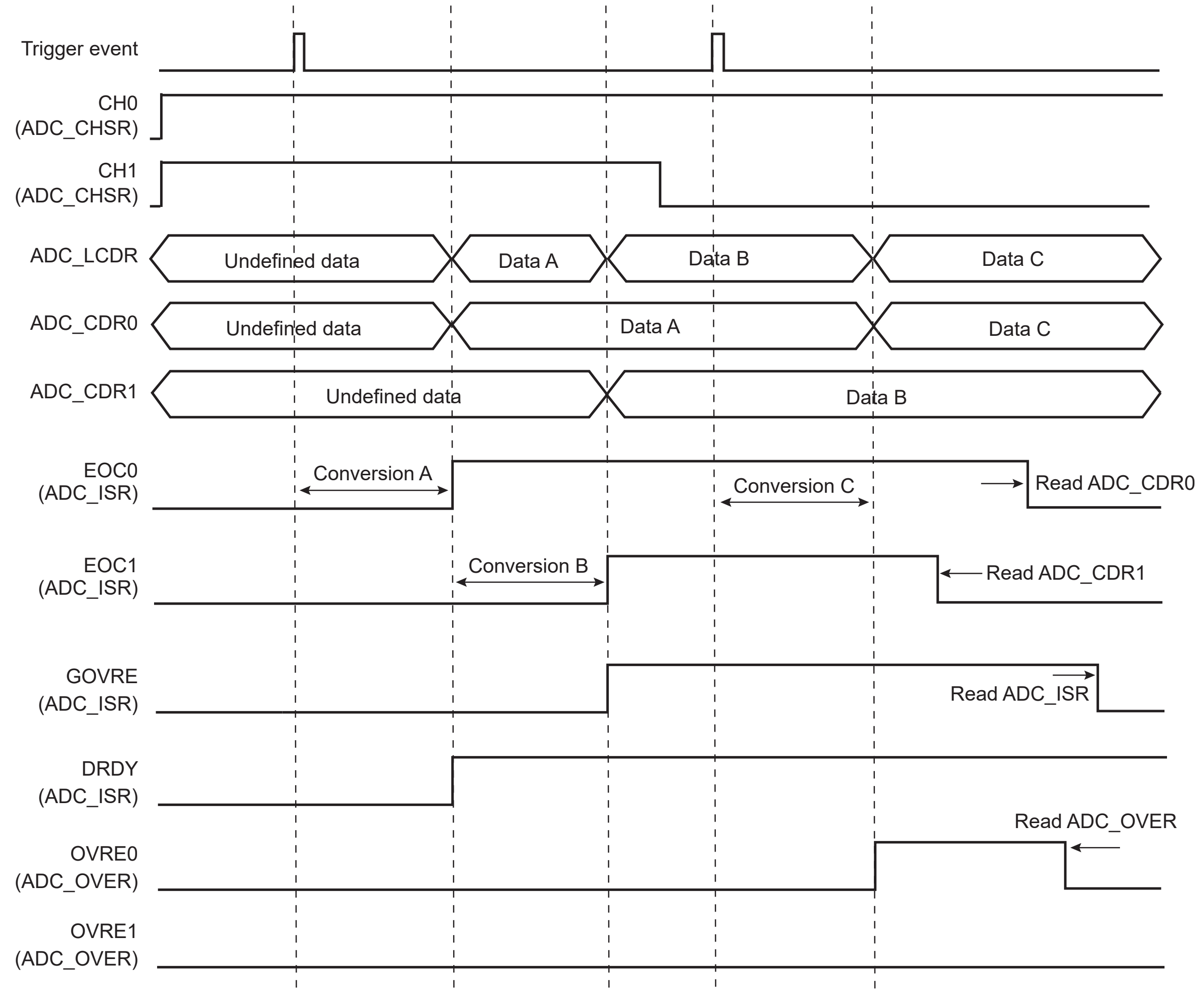


If ADC_CDRx is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status register (ADC_OVER).

If new data is converted when DRDY is high, ADC_ISR.GOVRE is set.

The OVREx flag is automatically cleared when ADC_OVER is read, and the GOVRE flag is automatically cleared when ADC_ISR is read.

Figure 38-5. EOCx, OVREx and GOVRE Flag Behavior



WARNING If the corresponding channel is disabled during a conversion or if it is disabled and then re-enabled during a conversion, its associated data and corresponding ADC_EOC_ISR.EOCx, ADC_ISR.GOVRE and ADC_OVER.OVREx flags are unpredictable.

38.5.6 Conversion Results Format

The conversion results can be signed (2's complement) or unsigned, depending on the value of ADC_EMR.SIGNMODE.

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to bit 15 (e.g., 0xF43 for 12-bit resolution is read as 0xFF43, and 0x467 is read as 0x0467).

38.5.7 Conversion Triggers

Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing the Control register (ADC_CR) with ADC_CR.START at 1.

The list of external/internal events is provided in ADC_MR. The hardware trigger is selected using ADC_MR.TRGSEL. The selected hardware trigger is enabled if TRGMOD = 1, 2 or 3 in the Trigger register (ADC_TRGR).

The ADC also provides a dual trigger mode (ADC_TEMPMR.TEMPON=1) in which the highest index channel can be sampled at a rhythm different from the other channels. The trigger of the last channel is generated by the RTC. See Temperature Sensor.

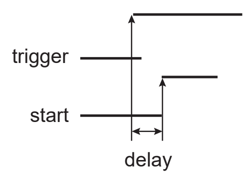
ADC_TRGR.TRGMOD selects the hardware trigger from the following:

- any edge, either rising or falling or both, detected on the external trigger pin ADTRG or internal triggers (provided by other peripherals)
- a continuous trigger, meaning the ADC Controller restarts the next sequence as soon as it finishes the current one
- a periodic trigger (generated by the ADC Controller), which is defined by programming ADC_TRGR.TRGPER

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence according to configuration of registers ADC_MR, ADC_SEQRx (Channel Sequence register), ADC_CHSR (Channel Status register).

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one ADC clock period. This delay introduces sampling jitter in the A/D conversion process and may therefore degrade the conversion performance (e.g., SNR, THD).

Figure 38-6. Hardware Trigger Delay



Only one start command is necessary to initiate a conversion sequence on all the enabled channels. The ADC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (ADC_CHER) and Channel Disable (ADC_CHDR) registers enable the analog channels to be enabled or disabled independently.

If the ADC is used with a DMA channel, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

38.5.8 Sleep Mode and Conversion Sequencer

The ADC Sleep mode maximizes power saving by automatically deactivating the ADC when it is not being used for conversions. Sleep mode is selected by setting ADC_MR.SLEEP.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at the lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the start-up period of the ADC. Refer to the section “Electrical Characteristics”.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a start-up time, the logic waits during this time and starts the conversion on the enabled channels. When all conversions are complete, the ADC is deactivated until the next trigger. Events triggered during the sequence are ignored.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences can be performed periodically using the internal timer (ADC_TRGR) or the PWM event line. The periodic acquisition of several samples can be processed automatically without any intervention of the processor via the DMA channel.

The sequence can be customized by programming ADC_SEQR1 and ADC_SEQR2 and setting ADC_MR.USEQ. This sequence is limited to 16 channels, from 0 to 15. The user can choose a specific order of channels and can program up to 16 conversions by sequence. The user is free to create a personal sequence by writing channel numbers in ADC_SEQR1 and ADC_SEQR2. Not only can channel numbers be written in any sequence, channel numbers can be repeated several times. When ADC_MR.USEQ is set, ADC_SEQR1.USCHx and ADC_SEQR2.USCHx are used to define the sequence. Only enabled USCHx fields will be part of the sequence. Each USCHx field has a corresponding enable, CHx-1, in ADC_CHER.

If 16 channels are used on an application board, there is no restriction of usage of the user sequence. However, if some ADC channels are not enabled for conversion but rather used as pure digital inputs, the respective indexes of these channels cannot be used in the user sequence fields (see ADC_SEQR1 and ADC_SEQR2). For example, if channel 4 is disabled (ADC_CHSR[4] = 0), ADC_SEQRx fields USCH1 up to USCH16 must not contain the value 4. Thus the length of the user sequence may be limited by this behavior.

As an example, if only four channels (CH0 up to CH3) are selected for ADC conversions, the user sequence length cannot exceed four channels. Each trigger event may launch up to four successive conversions of any combination of channels 0 up to 3, but no more (i.e., in this case the sequence CH0, CH0, CH1, CH1, CH1 is impossible).

A sequence that repeats the same channel several times requires more enabled channels than channels actually used for conversion. For example, the sequence CH0, CH0, CH1, CH1 requires four enabled channels (four free channels on application boards) whereas only CH0, CH1 are really converted.

Note: The reference voltage pins always remain connected in Normal mode as in Sleep mode.

38.5.9 Comparison Window

The ADC Controller features automatic comparison functions. It compares converted values to a low threshold, a high threshold or both, depending on the value of ADC_EMR.CMPMODE. The comparison can be done on all channels or only on the channel specified in ADC_EMR.CMPSEL. To compare all channels, ADC_EMR.CMPALL must be set.

If set, ADC_EMR.CMPTYPE can be used to discard all conversion results that do not match the comparison conditions. Once a conversion result matches the comparison conditions, all the subsequent conversion results are stored in ADC_LCDR (even if these results do not meet the comparison conditions). Setting ADC_CR.CMPRST immediately stops the conversion result storage until the next comparison match.

If ADC_EMR.CMPTYPE is cleared, all conversions are stored in ADC_LCDR. Only the conversions that match the comparison conditions trigger the ADC_ISR.COMPE flag.

Moreover, a filtering option can be set by writing the number of consecutive comparison matches needed to raise the flag. This number can be written and read in ADC_EMR.CMPFILTER. The filtering option is dedicated to reinforcing the detection of an analog signal exceeding a predefined threshold. The filter is cleared as soon as ADC_ISR is read, so this filtering function must be used with the DMA controller and works only when using Interrupt mode (no polling).

The flag can be read on ADC_ISR.COMPE and can trigger an interrupt.

The high threshold and the low threshold can be read/write in the Compare Window register (ADC_CWR).

Depending on the sign of the conversion, chosen with ADC_EMR.SIGNMODE, the high threshold and low threshold values must be signed or unsigned to maintain consistency during the comparison. If the conversion is signed, both thresholds must also be signed; if the conversion is unsigned, both thresholds must be unsigned. If comparison occurs on all channels, SIGNMODE must be set to ALL_UNSIGNED or ALL_SIGNED and the thresholds must be set accordingly.

38.5.10 Differential and Single-ended Input Modes

38.5.10.1 Input-Output Transfer Functions

The ADC can be configured to operate in the following input voltage modes:

- Single-ended—ADC_CCR.DIFFx = 0. This is the default mode after a reset.
- Differential—ADC_CCR.DIFFx = 1 (see the figure Analog Full Scale Ranges in Single-ended/Differential Applications). In Differential mode, the ADC requires differential input signals with a VDD/2 common mode voltage (refer to the section “Electrical Characteristics”).

The following equations give the unsigned ADC input-output transfer function in each mode (see Note). With signed conversions (see ADC_EMR.SIGNMODE), subtract 2047 from the ADC_LCDR.LDATA value given below.

In the formulae below, REFP = ADVREFP, REFN = GNDIN33.

Single-ended mode:

$$\mathrm{ADC\_LCDR.LDATA} = \frac{\mathrm{ADx} - \mathrm{REFN}}{\mathrm{REFP} - \mathrm{REFN}} \times 2^{12}$$

Differential mode:

$$\text{ADC\_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{ADx+1}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

Note: Equations assume ADC_EMR.OSR = 1

If ADC_MR.ANACH is set, the ADC can manage both differential channels and single-ended channels. If ADC_MR.ANACH is cleared, the parameters defined in ADC_CCR are applied to all channels.

The following table gives the internal positive and negative ADC inputs assignment with respect to the programmed mode (ADC_CCR.DIFFx).

For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

Figure 38-7. Analog Full Scale Ranges in Single-ended/Differential Applications

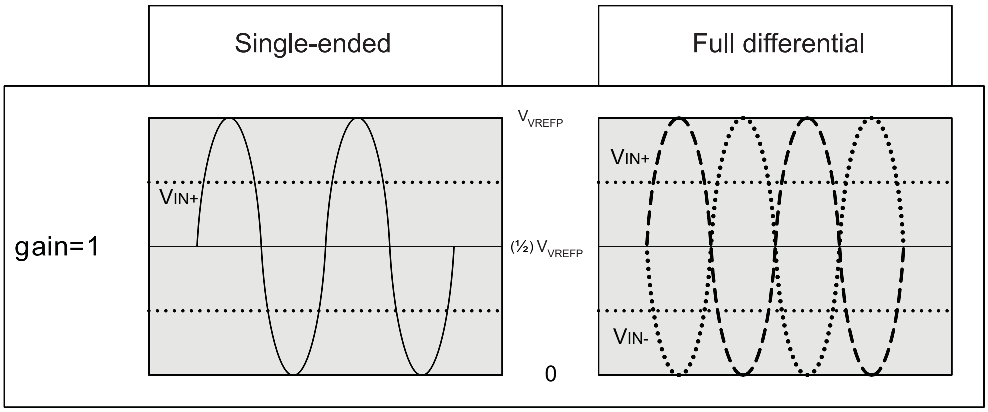


Table 38-1. Input Pins and Channel Numbers in Single-ended and Differential Modes

Internal ADC Inputs (VIN+, VIN-)		Channel Numbers	
Single-ended Mode	**Differential Mode**	**Single-ended Mode**	**Differential Mode**
AD0, GNDIN33	AD0, AD1	CH0	CH0
AD1, GNDIN33	–	CH1	
AD2, GNDIN33	AD2, AD3	CH2	CH2
AD3, GNDIN33	–	CH3	
AD4, GNDIN33	AD4, AD5	CH4	CH4
AD5, GNDIN33	–	CH5	
AD6, GNDIN33	AD6, AD7	CH6	CH6
AD7, GNDIN33	–	CH7	
AD8, GNDIN33	AD8, AD9	CH8	CH8
AD9, GNDIN33	–	CH9	
AD10, GNDIN33	AD10, AD11	CH10	CH10
AD11, GNDIN33	–	CH11	
AD12, GNDIN33	AD12, AD13	CH12	CH12
AD13, GNDIN33	–	CH13	
AD14, GNDIN33	AD14, AD15	CH14	CH14
AD15, GNDIN33	–	CH15	

...........continued

Internal ADC Inputs (VIN+, VIN-)		Channel Numbers	
Single-ended Mode	Differential Mode	Single-ended Mode	Differential Mode
–	–	Not Available	Not Available
–	–	Not Available	

38.5.10.2 Alternate Multiplexer Selection (Safety)

The ADC Controller features an alternate input multiplexer that can be used to test the main input multiplexer in the context of safety-critical applications.

The alternate input multiplexer can be selected by enabling ADC_EMR.ALTCH. When selecting the alternate multiplexer, ADC_MR.TRACKTIM must be set to 15 and ADC_EMR.TRACKX must be set to 1.

38.5.11 ADC Timings

The ADC start-up time is programmed using ADC_MR.STARTUP. Refer to the section “Electrical Characteristics”.

The ADC Controller provides an inherent tracking time of six ADC clock cycles.

A minimal tracking time is necessary for the ADC to ensure the best converted final value between two conversions. The tracking time can be adjusted to accommodate a range of source impedances. If more than six ADC clock cycles are required, the tracking time can be increased by using ADC_MR.TRACKTIM and ADC_EMR.TRACKX.

⚠WARNING No input buffer amplifier to isolate the source is included in the ADC. Refer to the section “Electrical Characteristics”.

38.5.12 Temperature Sensor

The temperature sensor is internally connected to the channel with the highest index. For temperature measurement, ADC_TEMPMR.TEMPON must be set.

Temperature measurement can be performed in different ways through the ADC Controller. The different methods of measurement depend on configuration bits ADC_TRGR.TRGMOD and ADC_CHSR.CHmax (max=highest index).

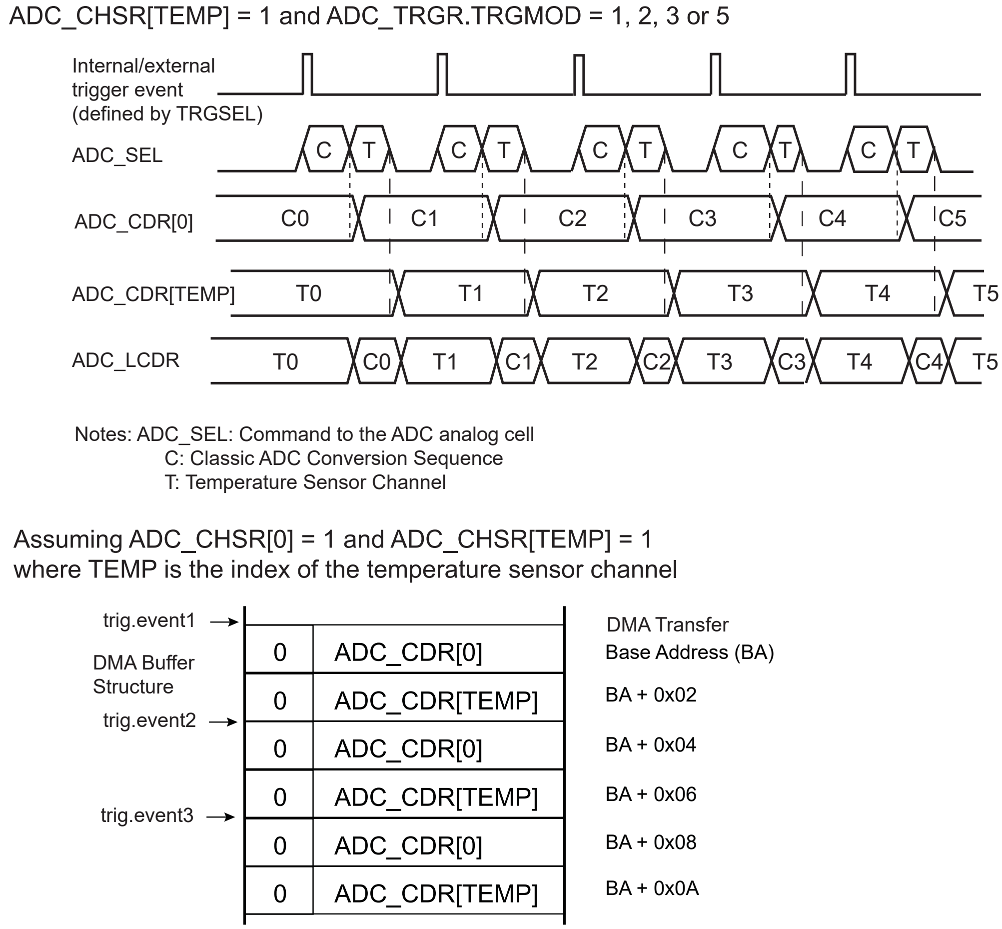
Temperature measurement can be triggered like the other channels by enabling its associated conversion channel, writing 1 in ADC_CHER.CHmax (max=highest index).

The manual start can only be performed if ADC_TRGR.TRGMOD = 0. When ADC_CR.START is set, the temperature sensor channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in the ADC_CDRmax (max=highest index) register and the associated ADC_EOC_ISR.EOCmax (max=highest index) flag is set.

If ADC_TRGR.TRGMOD = 1, 2, 3 or 5, the temperature sensor channel is periodically converted together with other enabled channels and the result is placed on the ADC_LCDR and ADC_CDRmax (max=highest index) registers. Thus, the temperature conversion result is part of the DMA Controller buffer. The temperature channel can be enabled/disabled at any time but this may not be optimal for downstream processing.

When the conversion result matches the conditions defined in the ADC_TEMPMR and ADC_TEMPCWR, the ADC_ISR.TEMPCHG flag is set.

Figure 38-8. Non-optimized Temperature Conversion



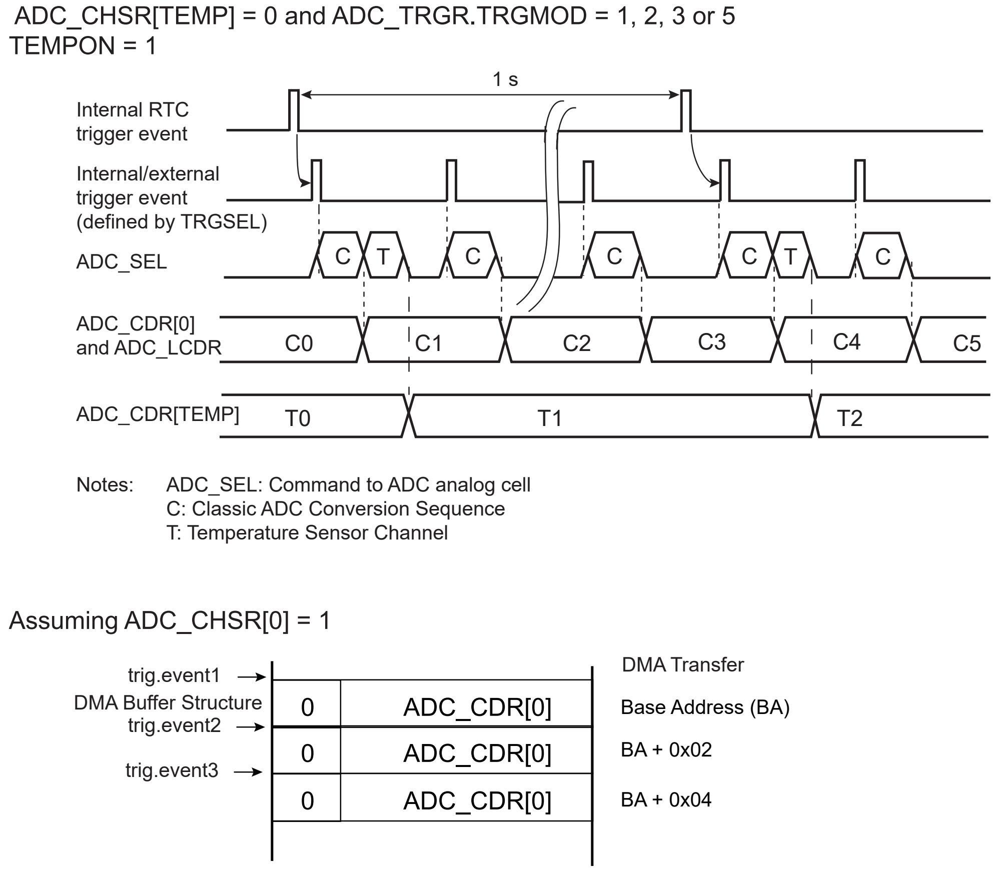
The temperature factor having a slow variation rate and being potentially totally different from the other conversion channels, the ADC Controller allows a different way of triggering the measure when ADC_TEMPMR.TEMPON is set but ADC_CHSR.CHmax (max=highest index) is not set.

Under these conditions, the measure is triggered every second by means of an internal trigger generated by the RTC, always enabled and totally independent of the internal/external triggers. The RTC event will be processed on the next internal/external trigger event, as described in the following figure. The internal/external trigger is selected through ADC_MR.TRGSEL.

In this mode of operation, the temperature sensor is only powered for a period of time covering the start-up time and conversion time (see the figure Temperature Conversion Only for more details).

Every second, a conversion is scheduled for the temperature channel but the result of the conversion is only uploaded in the ADC_CDRmax (max=highest index) register and not in ADC_LCDR. Therefore, there is no change in the structure of the DMA Controller buffer due to the conversion of the temperature channel, only the enabled channels are kept in the buffer. The end of conversion of the temperature channel is reported by the ADC_EOC_ISR.EOCmax (max=highest index) flag.

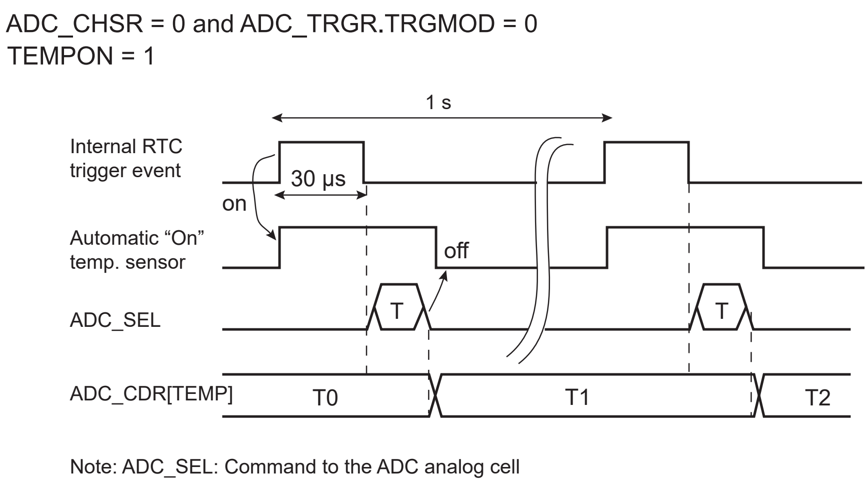
Figure 38-9. Optimized Temperature Conversion Combined With Classical Conversions



If TEMPON = 1, TRGMOD = 0 and none of the channels are enabled in ADC_CHSR (ADC_CHSR = 0), then only the temperature channel is converted at a rate of 1 conversion per second (see the figure Temperature Conversion Only).

This mode of operation, when combined with the Sleep mode operation of the ADC Controller, provides a low-power mode for temperature measurement (assuming there is no other ADC conversion to schedule at a high sampling rate, or simply no other channel to convert).

Figure 38-10. Temperature Conversion Only



Note: ADC_SEL: Command to the ADC analog cell

Furthermore, it is possible to raise a flag only if there is a predefined change in the temperature measure. The user can define a range of temperatures or a threshold in ADC_TEMPCWR, and the mode of comparison that can be programmed in ADC_TEMPMR.TEMPCMPMOD. These values define how the ADC_ISR.TEMPCHG flag is raised.

In any case, if TEMPON is set, the temperature can be read at anytime in ADC_CDRmax (max=highest index) without any specific software intervention.

38.5.13 Enhanced Resolution Mode and Digital Averaging Function

38.5.13.1 Enhanced Resolution Mode

The Enhanced Resolution mode is enabled if ADC_EMR.OSR is configured to 1, 2, 3 or 4. The enhancement is based on a digital averaging function.

There is no averaging on the last index channel if the measure is triggered by an RTC event.

In this mode, the ADC Controller will trade off conversion speed against accuracy by averaging multiple samples, thus providing a digital low-pass filter function.

The selected oversampling ratio applies to all enabled channels when triggered by an RTC event. The following formula applies:

$$\mathrm{ADC\_LCDR.LDATA} = \frac{1}{M} \times \sum_{k=0}^{k=N-1} ADC(k)$$

where N and M are given in the table below.

Table 38-2. Digital Averaging Function Configuration Versus OSR Values

ADC_EMR.OSR Value	ADC_LCDR.LDATA Length	N Value	M Value	Full Scale Value	Maximum Value
0	12 bits	1	1	4095	4095
1	13 bits	4	2	8191	8190
2	14 bits	16	4	16383	16381
3	15 bits	64	8	32767	32766
4	16 bits	256	16	65535	65533

The average result is valid in ADC_CDRx (x = channel index) only if the ADC_EOC_ISR.EOCn flag is set and if the ADC_OVER.OVREn flag is cleared. The average result for all channels is valid in ADC_LCDR only if ADC_ISR.DRDY is set and ADC_ISR.GOVRE is cleared.

Note that ADC_CDRs are not buffered. Therefore, when an averaging sequence is ongoing, the value in these registers changes after each averaging sample. However, overrun flags in ADC_OVER rise as soon as the first sample of an averaging sequence is received. Thus, the previous averaged value is not read, even if the new averaged value is not ready.

Consequently, when an overrun flag rises in ADC_OVER, it means that the previous unread data is lost but it does not mean that this data has been overwritten by the new averaged value, as the averaging sequence concerning this channel can still be ongoing.

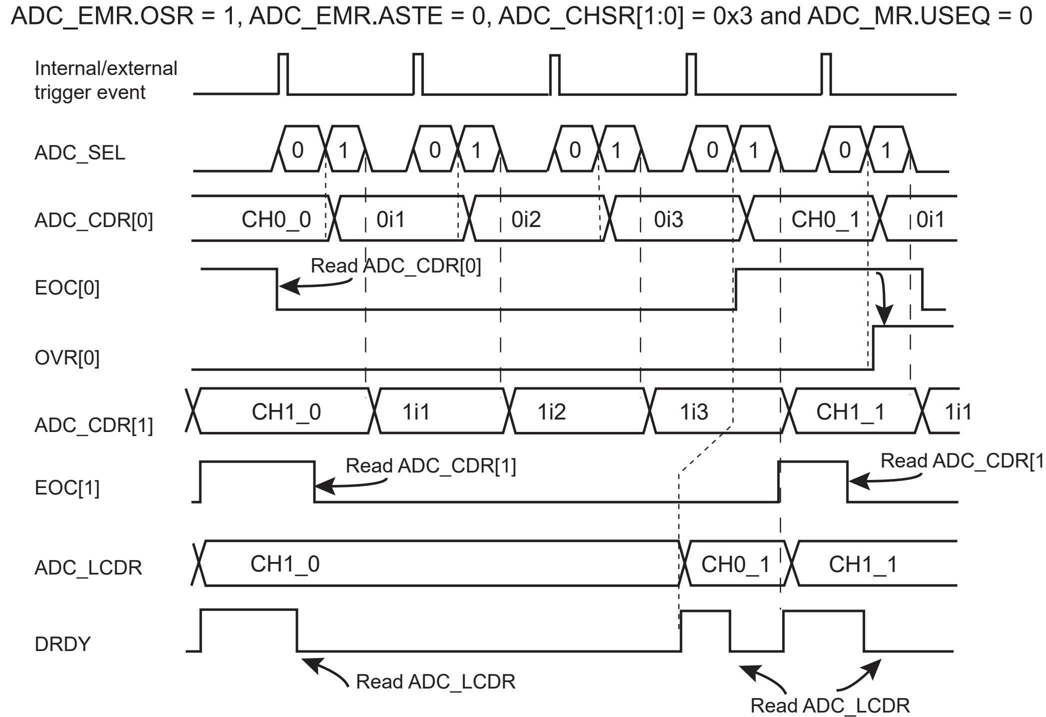
When an oversampling is performed, the maximum value that can be read on ADC_CDRx or ADC_LCDR is not the full-scale value, even if the maximum voltage is supplied on the analog input. See the table Digital Averaging Function Configuration Versus OSR Values.

38.5.13.2 Averaging Function Versus Trigger Events

The samples can be defined in different ways for the averaging function, depending on the configuration of ADC_EMR.ASTE and ADC_MR.USEQ.

When ADC_MR.USEQ = 0, there are two possible ways to generate the averaging through the trigger event. If ADC_EMR.ASTE = 0, every trigger event generates one sample for each enabled channel, as described in the following figure. Therefore, four trigger events are requested to obtain the result of averaging if ADC_EMR.OSR = 1.

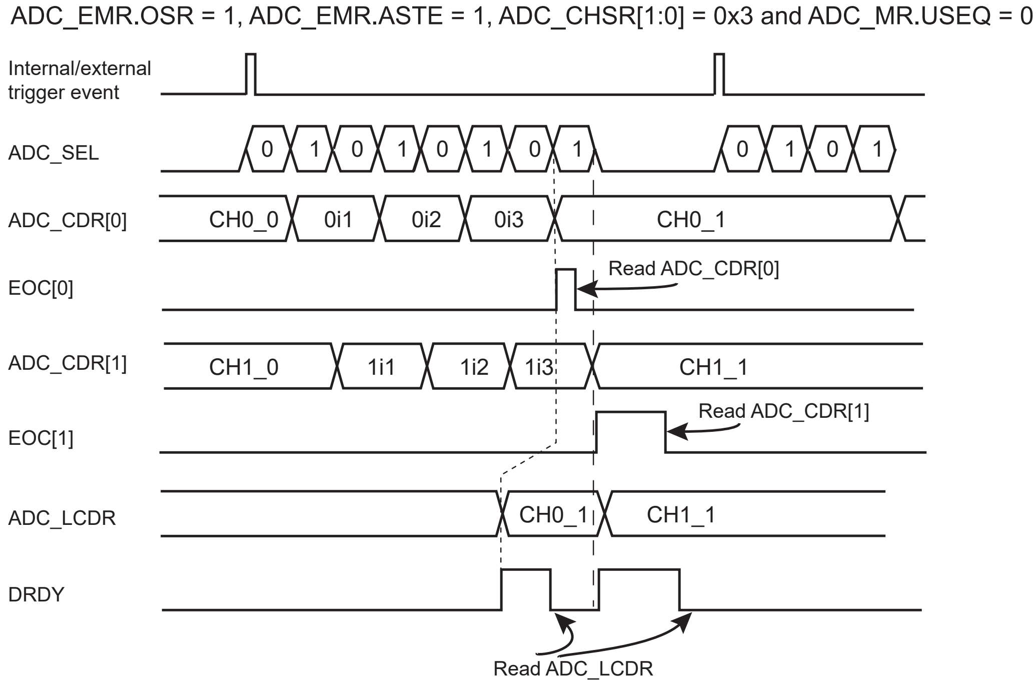
Figure 38-11. Digital Averaging Function Waveforms Over Multiple Trigger Events



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

If ADC_EMR.ASTE = 1 and ADC_MR.USEQ = 0, the sequence to be converted, defined in ADC_CHSR, is automatically repeated n times (where n corresponds to the oversampling ratio defined in ADC_EMR.OSR). As a result, only one trigger is required to obtain the result of the averaging function, as described in the following figure.

Figure 38-12. Digital Averaging Function Waveforms on a Single Trigger Event



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

When USEQ = 1, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion, as described in the following figure.

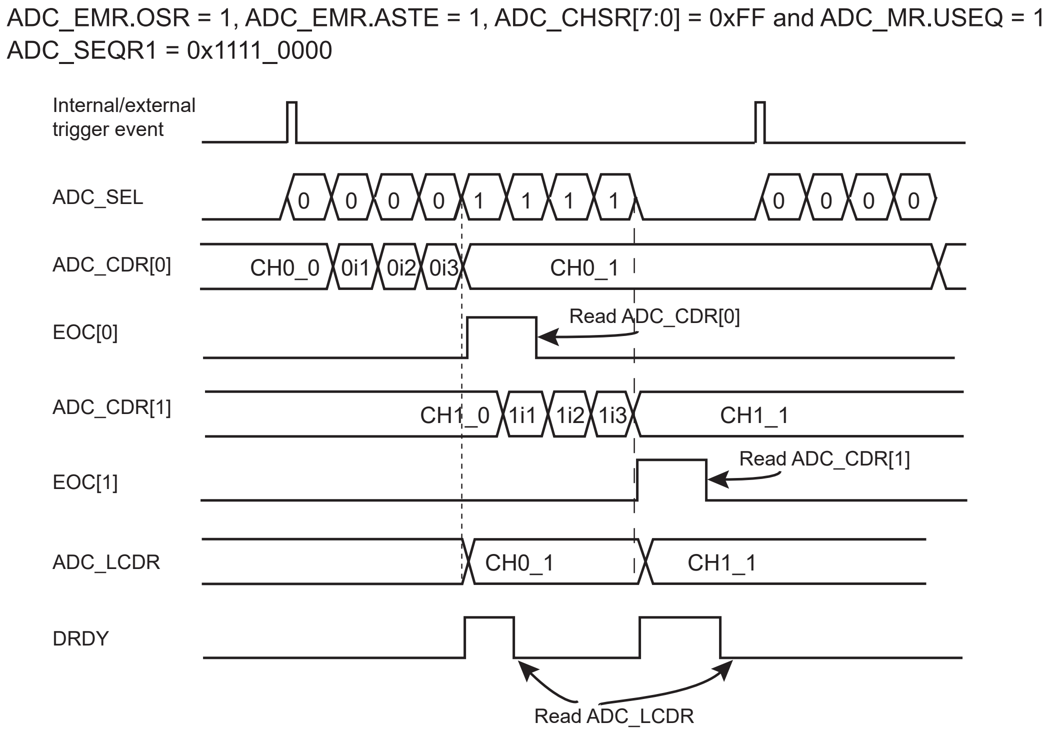
When USEQ = 1 and ASTE = 1, OSR can be only configured to 1. Up to three channels can be converted in this mode. The averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event. The ADC real sample rate remains the maximum ADC sample rate divided by 4.

It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

Table 38-3. Example Sequence Configurations (USEQ = 1, ASTE = 1, OSR = 1)

Register	Number of Channels Non-interleaved Averaging - Register Value		
	1 (e.g., CH0)	2 (e.g., CH0, CH1)	3 (e.g., CH0, CH1, CH2)
ADC_CHSR	0x0000000F	0x000000FF	0x00000FFF
ADC_SEQR1	0x00000000	0x11110000	0x11110000
ADC_SEQR2	0x00000000	0x00000000	0x00002222

Figure 38-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

38.5.14 Automatic Error Correction

The ADC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are the same for all channelsprogrammable per channel.

To enable error correction, the corresponding ECORRx bit must be set in the Channel Error Correction register (ADC_CECR). The offset and gain values used to compensate the results are the same for all correction-enabled channels and programmed in the Correction Values register (ADC_CVR) set on a ‘per channel’ basis using the Correction Select register (ADC_COSR) and the Correction Values register (ADC_CVR). ADC_COSR is used to select the channel to be displayed in ADC_CVR. This selection applies to both the read and the write operations in ADC_CVR.

ADC_EMR.ADCMODE is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see ADC_EMR). ADCMODE uses two internal references (ADVREFP, GNDIN33) to be measured and to extract the offset and gain error from 3 point-measurement codes. If some references already exist on the final application connected to some input channel ADx, they can be used as a replacement of the ADCMODE to generate the 2 or 3 points of calibration and used to extract the GAINCORR and OFFSETCORR.

After a reset, the ADC running mode is Normal mode. Offset Error mode and Gain Error mode are used to determine values of offset compensation and gain compensation, respectively, to apply to conversion results. The following table provides formulas to obtain the compensation values, with:

- OFFSETCORR—the Offset Correction value. OFFSETCORR is a signed value.
- GAINCORR—the Gain Correction value
- GCi—the intermediate Gain Compensation value
- Gs—the value 13
- ConvValue—the value converted by the ADC (as returned in ADC_LCDR or ADC_CDR)
- Resolution—the resolution used to process the conversion (either RESOLUTION, RESOLUTION+1 or RESOLUTION+2).

Table 38-4. ADC Running Modes

ADC_EMR.ADCMODE	Mode	Description
0	Normal	Normal mode of operation to perform conversions
1	Offset Error	For unsigned conversions: OFFSETCORR = ConvValue − $2^{(\text{Resolution} - 1)}$
		For signed conversions: OFFSETCORR = ConvValue
2	Gain Error	GCi = ConvValue
3		$\text{GAINCORR} = \frac{3584}{\text{GCi} - \text{ConvValue}} \times 2^{(\text{Gs})}$

The final conversion result after error correction is obtained using the following formula:

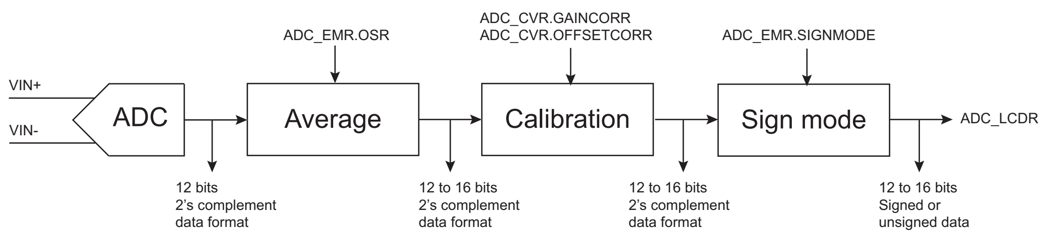
$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(\text{Gs})}}$$

38.5.14.1 Calibration Mode of Operation

The ADC is built to convert positive and negative voltages around a common mode voltage (ADVREFP-GNDIN33)/2. This is why the ADC output, averaging and calibration are performed as 2's complements. An ADC positive integer output is generated only in the last stage to generate unsigned data.

The ADC code is first averaged when requested; calibration is applied afterward. See the following figure.

Figure 38-14. ADC Signal Processing



Three internal reference voltages, VM, VL and VH, are generated from ADVREFP, with:

- VL = (ADVREFP-GNDIN33)x(1/16)
- VH = (ADVREFP-GNDIN33) x(15/16)
- VM = (ADVREFP-GNDIN33)/2

In Single mode non-signed outputs, the references are ideally converted as:

- VL_code = 256
- VH_code = 3840
- VM_code = 2048

In Differential mode signed outputs, the references are ideally converted as:

- (VL-VH)_code = -1792
- (VH-VL)_code = +1792
- VM_code = 0

The difference between the two voltage reference codes is the value 3584.

For calibration, three values are measured using ADCMODE 1,2,3.

In the table ADC Running Modes, ConvValue is the current converted value.

38.5.14.1.1 First Conversion in ADCMODE=1

The ADC converted value is placed in the OFFSETCORR field (even after averaging) in a Sign mode signed output. If the output is unsigned, subtract half the code dynamic, which depends on the resolution (averaging).

OFFSETCORR is the offset at mid-range in an unsigned configuration.

38.5.14.1.2 Second Conversion in ADCMODE=2

This gives an upper code for gain calculation:

ADC output VH_Code = GCi = ConvValue

38.5.14.1.3 Third Conversion in ADCMODE=3

This gives the lowest code for the gain calculation:

ADC output VL_code = new ConvValue

Finally, VH_Code-VL_Code = GCI-ConvValue is the spread between the two codes, ideally at 3584 when there is no gain error.

38.5.14.1.4 Final GAINCORR Computation

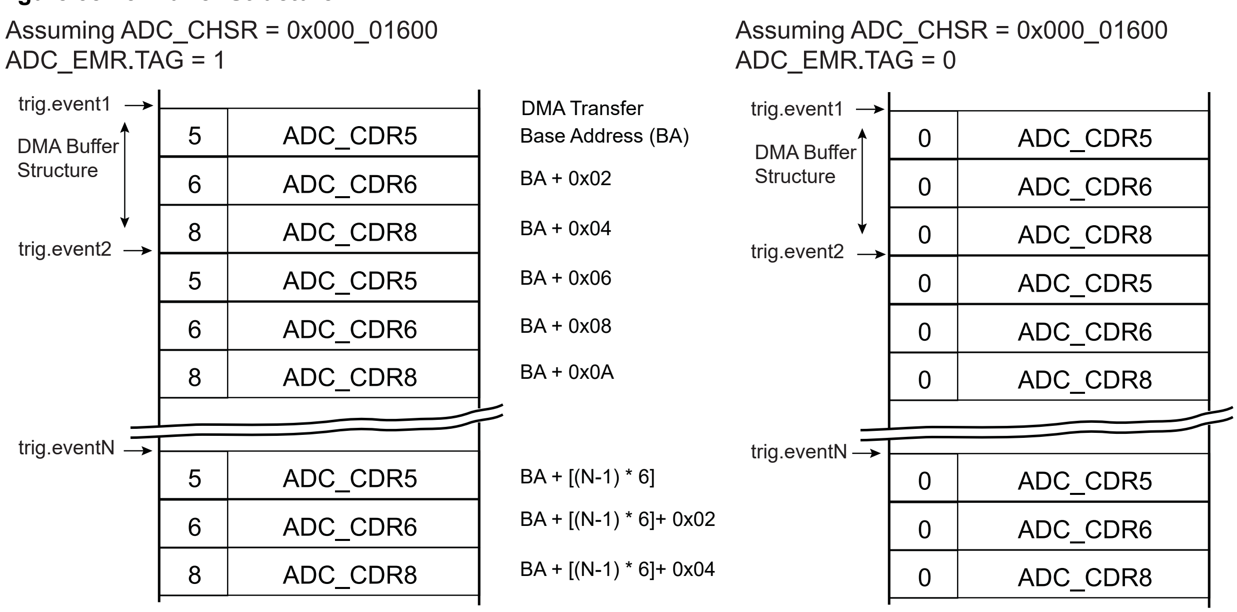
The final formula to compute GAINCORR can be applied. This formula is given without averaging (ADC resolution at 12 bits).

If an averaging is applied to get the ADC code in ADCMODE 2 and 3, then it is needed to scale down the output code to fit a 12-bit resolution.

38.5.15 Buffer Structure without FIFO

The DMA read channel is triggered when ADC_FMR.ENFIFO is set to 0 (see ENFIFO: Enable FIFO) and each time a new data is stored in ADC_LCDR. The same structure of data is repeatedly stored in ADC_LCDR each time a trigger event occurs. Depending on the user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_SEQR2, ADC_TSMR), the structure differs. Each data read to DMA buffer, carried on a half-word (16 bits), consists of last converted data right-aligned and when TAG is set in ADC_EMR, the four most significant bits carry the channel number, thus allowing an easier post-processing in the DMA buffer or a better checking of the DMA buffer integrity.

Figure 38-15. Buffer Structure



.

38.5.15.1 Classic ADC Channels Only

The structure of data within the buffer is defined by ADC_MR, ADC_CHSR, ADC_SEQRx. See the figure Buffer Structure.

If the user sequence is not used (i.e., ADC_MR.USEQ is cleared), then only the value of ADC_CHSR defines the data structure. For each trigger event, enabled channels are consecutively stored in ADC_LCDR and automatically read to the buffer.

When the user sequence is configured (i.e., ADC_MR.USEQ is set), not only does ADC_CHSR modify the data structure of the buffer, but ADC_SEQRx registers may modify the data structure of the buffer as well.

38.5.16 Buffer Structure with FIFO

The DMA read channel is triggered when the fields ENFIFO, ENLEVEL and CHUNK are used (see ADC_FMR).

If the configuration ENFIFO is high and ENLEVEL is low, the DMA read channel is triggered as soon as one data is written.

When at least one data is ready in the FIFO, the RXRDY flag rises and remains high as long as there is at least one data to be read in the FIFO. When the entire FIFO has been filled with data, the RXFULL flag rises. If a new data is written into the RX FIFO while RXFULL is high, then the RXOVR flag rises. This flag remains high until ADC_ISR is read.

Once all data contained in the RX FIFO have been read, the RXEMPTY flag rises. If a data is read while the RX FIFO is empty, an underrun occurs and the RXUDR flag rises. This flag remains high until ADC_ISR is read.

Once a data is written in the RX FIFO, the RXRDY flag rises. If the corresponding interrupt has been enabled, an interrupt is generated and remains high as long as a data is available in the RX FIFO.

The FIFO features chunk management. When the number of filled spaces reaches the chunk size defined by ADC_FMR.CHUNK, the DMA read channel is triggered and the RXCHUNK flag is raised, ensuring that chunk size data can be read consecutively. This flag is reset once CHUNK data are read. This chunk definition can be used with the DMA to define how many accesses can be performed. In this case, the chosen chunk size must match the chunk value defined in the DMA.

The DMA read channel is triggered when a new data is stored in FIFO. The same structure of data is repeatedly stored in FIFO each time a trigger event occurs. Depending on the user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_SEQR2, ADC_TSMR), the structure differs. Each data read to DMA buffer, carried on a half-word (16 bits), consists of last converted data right-aligned, and when ADC_EMR.TAG is set, the four most significant bits carry the channel number, thus allowing an easier post-processing in the DMA buffer or a better checking of the DMA buffer integrity.

ADC_FMR.FIFOCNT gives the number of conversions available in the FIFO.

38.5.17 Fault Event

The ADC Controller internal fault output is directly connected to the PWM fault input. The fault event may be asserted depending on the configuration of comparison registers and converted values.

Two types of comparison trigger a fault event sent to the PWM:

- The first comparison type is based on ADC_CWR settings, i.e., on all converted channels except the last one;
- The second comparison type is linked to the last channel (ADC_TEMPCWR settings) where the temperature is measured.

As an example, overcurrent and temperature values exceeding user-defined limits trigger a fault to the PWM.

When the comparison event occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus, the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

38.5.18 Register Write Protection

To prevent any single software error from corrupting ADC behavior, certain registers in the address space can be write-protected by setting the bits WPEN and WPITEN in the ADC Write Protection Mode Register (ADC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the ADC Write Protection Status Register (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading ADC_WPSR.

The following registers are write-protected when ADC_WPMR.WPEN is set:

- ADC Mode Register
- ADC Channel Sequence Register 1
- ADC Channel Sequence Register 2

- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Temperature Sensor Mode Register
- ADC Temperature Compare Window Register
- ADC Extended Mode Register
- ADC FIFO Mode Register
- ADC Compare Window Register
- ADC Channel Configuration Register
- ADC Analog Control Register
- ADC Trigger Register
- ADC Correction Select Register
- ADC Correction Values Register
- ADC Channel Error Correction Register

The following registers are write-protected when ADC_WPMR.WPITEN is set:

- ADC Interrupt Enable Register
- ADC Interrupt Disable Register

The following register is write-protected when ADC_WPMR.WPCREN is set:

- ADC Control Register

38.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ADC_CR	31:24								
		23:16								
		15:8								
		7:0				CMPRST	SWFIFO		START	SWRST
0x04	ADC_MR	31:24	USEQ	ALWAYS1	TRANSFER[1:0]		TRACKTIM[3:0]			
		23:16	ANACH				STARTUP[3:0]			
		15:8	PRESCAL[7:0]							
		7:0		FWUP	SLEEP		TRGSEL[2:0]			
0x08	ADC_SEQR1	31:24	USCH8[3:0]				USCH7[3:0]			
		23:16	USCH6[3:0]				USCH5[3:0]			
		15:8	USCH4[3:0]				USCH3[3:0]			
		7:0	USCH2[3:0]				USCH1[3:0]			
0x0C	ADC_SEQR2	31:24	USCH16[3:0]				USCH15[3:0]			
		23:16	USCH14[3:0]				USCH13[3:0]			
		15:8	USCH12[3:0]				USCH11[3:0]			
		7:0	USCH10[3:0]				USCH9[3:0]			
0x10	ADC_CHER	31:24	CH31	CH30						
		23:16								
		15:8	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x14	ADC_CHDR	31:24	CH31	CH30						
		23:16								
		15:8	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x18	ADC_CHSR	31:24	CH31	CH30						
		23:16								
		15:8	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x1C ... 0x1F	Reserved									
0x20	ADC_LCDR	31:24				CHNBOSR[4:0]				
		23:16								
		15:8	LDATA[15:8]							
		7:0	LDATA[7:0]							
0x20	ADC_LCDR (NO_OSR)	31:24								
		23:16								
		15:8	NO_OSR_CHNBOSR[3:0]				NO_OSR_LDATA[11:8]			
		7:0	NO_OSR_LDATA[7:0]							
0x24	ADC_IER	31:24						COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x28	ADC_IDR	31:24						COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x2C	ADC_IMR	31:24						COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x30	ADC_ISR	31:24						COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	ADC_EOC_IER	31:24	EOC31	EOC30						
		23:16								
		15:8	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x38	ADC_EOC_IDR	31:24	EOC31	EOC30						
		23:16								
		15:8	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x3C	ADC_EOC_IMR	31:24	EOC31	EOC30						
		23:16								
		15:8	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x40	ADC_EOC_ISR	31:24	EOC31	EOC30						
		23:16								
		15:8	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x44	ADC_TEMPMR	31:24								
		23:16								
		15:8								
		7:0			TEMPCMPMOD[1:0]					TEMPON
0x48	ADC_TEMPCWR	31:24					THIGHTHRES[11:8]			
		23:16	THIGHTHRES[7:0]							
		15:8					TLOWTHRES[11:8]			
		7:0	TLOWTHRES[7:0]							
0x4C	ADC_OVER	31:24	OVRE31	OVRE30						
		23:16								
		15:8	OVRE15	OVRE14	OVRE13	OVRE12	OVRE11	OVRE10	OVRE9	OVRE8
		7:0	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
0x50	ADC_EMR	31:24		ALTCH	ADCMODE[1:0]			SIGNMODE[1:0]		TAG
		23:16	TRACKX[1:0]		ALWAYS0	ASTE		OSR[2:0]		
		15:8			CMPFILTER[1:0]				CMPALL	CMPSEL[4]
		7:0	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
0x54	ADC_CWR	31:24	HIGHTHRES[15:8]							
		23:16	HIGHTHRES[7:0]							
		15:8	LOWTHRES[15:8]							
		7:0	LOWTHRES[7:0]							
0x58 ... 0x5B	Reserved									
0x5C	ADC_CCR	31:24								
		23:16								
		15:8	DIFF15	DIFF14	DIFF13	DIFF12	DIFF11	DIFF10	DIFF9	DIFF8
		7:0	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
0x60	ADC_CDR0	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
...										
0xDC	ADC_CDR31	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0xE0	ADC_ACR	31:24								
		23:16								SRCLCH
		15:8								
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE4	ADC_FMR	31:24								
		23:16	FIFOCNT[7:0]							
		15:8								
		7:0	CHUNK[3:0]						ENLEVEL	ENFIFO
0xE8 ... 0xFF	Reserved									
0x0100	ADC_TRGR	31:24	TRGPER[23:16]							
		23:16	TRGPER[15:8]							
		15:8	TRGPER[7:0]							
		7:0						TRGMOD[2:0]		
0x0104	ADC_COSR	31:24								
		23:16								
		15:8								
		7:0				CSEL[4:0]				
0x0108	ADC_CVR	31:24	GAINCORR[15:8]							
		23:16	GAINCORR[7:0]							
		15:8	OFFSETCORR[15:8]							
		7:0	OFFSETCORR[7:0]							
0x010C	ADC_CECR	31:24	ECORR31	ECORR30						
		23:16								
		15:8	ECORR15	ECORR14	ECORR13	ECORR12	ECORR11	ECORR10	ECORR9	ECORR8
		7:0	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
0x0110 ... 0x0117	Reserved									
0x0118	ADC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x011C	ADC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

38.6.1 ADC Control Register

Name: ADC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CMPRST	SWFIFO		START	SWRST
Access				W	W		W	W
Reset				–	–		–	–

Bit 4 – CMPRST Comparison Restart

Value	Description
0	No effect.
1	Stops the conversion result storage until the next comparison match.

Bit 3 – SWFIFO Software FIFO Reset

Value	Description
0	No effect.
1	Resets the internal FIFO, simulating a hardware reset.

Bit 1 – START Start Conversion

Value	Description
0	No effect.
1	Begins analog-to-digital conversion.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the ADC, simulating a hardware reset.

38.6.2 ADC Mode Register

Name: ADC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USEQ	ALWAYS1	TRANSFER[1:0]		TRACKTIM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ANACH				STARTUP[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PRESCAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		FWUP	SLEEP		TRGSEL[2:0]			
Access		R/W	R/W		R/W	R/W	R/W	
Reset		0	0		0	0	0	

Bit 31 – USEQ User Sequence Enable

Value	Name	Description
0	NUM_ORDER	Normal mode: The controller converts channels in a simple numeric order depending only on the channel index.
1	REG_ORDER	User Sequence mode: The sequence respects what is defined in ADC_SEQR1 and ADC_SEQR2 and can be used to convert the same channel several times.

Bit 30 – ALWAYS1 Must always be written to 1

Bits 29:28 – TRANSFER[1:0] Transfer Time
Must be set to 2 to ensure the optimal transfer time.

Bits 27:24 – TRACKTIM[3:0] Tracking Time

ADC_EMR.TRACKX	TRACKTIM		
	0 to 3	4 to 14	15
0	$6 \times t_{ADCCLK}$		$7 \times t_{ADCCLK}$
1	$6 \times t_{ADCCLK}$	$([4 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	
2	Not Applicable	$([8 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	
3	Not Applicable	$([16\times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	

Bit 23 – ANACH Analog Change

Value	Name	Description
0	NONE	No analog change on channel switching: DIFF0 is used for all channels.
1	ALLOWED	Allows different analog settings for each channel. See ADC_CCR.

Bits 19:16 – STARTUP[3:0] Start-Up Time

Value	Name	Description
0	SUT0	0 period of ADCCLK
1	SUT8	8 periods of ADCCLK
2	SUT16	16 periods of ADCCLK
3	SUT24	24 periods of ADCCLK
4	SUT64	64 periods of ADCCLK
5	SUT80	80 periods of ADCCLK
6	SUT96	96 periods of ADCCLK
7	SUT112	112 periods of ADCCLK
8	SUT512	512 periods of ADCCLK
9	SUT576	576 periods of ADCCLK
10	SUT640	640 periods of ADCCLK
11	SUT704	704 periods of ADCCLK
12	SUT768	768 periods of ADCCLK
13	SUT832	832 periods of ADCCLK
14	SUT896	896 periods of ADCCLK
15	SUT960	960 periods of ADCCLK

Bits 15:8 – PRESCAL[7:0] Prescaler Rate Selection
PRESCAL = ($f_{peripheral\ clock}$ / (2 × f_{ADCCLK})) – 1.

Bit 6 – FWUP Fast Wakeup

Value	Name	Description
0	OFF	If SLEEP is 1, then both ADC core and reference voltage circuitry are OFF between conversions.
1	ON	If SLEEP is 1, then Fast Wake-up Sleep mode: The voltage reference is ON between conversions and ADC core is OFF.

Bit 5 – SLEEP Sleep Mode

Value	Name	Description
0	NORMAL	Normal mode: The ADC core and reference voltage circuitry are kept ON between conversions.
1	SLEEP	Sleep mode: The wake-up time can be modified by programming the FWUP bit.

Bits 3:1 – TRGSEL[2:0] Trigger Selection
The trigger selection can be performed only if ADC_TRGR.TRGMOD = 1, 2 or 3.

Value	Name	Description
0	ADC_TRIG0	ADTRG
1	ADC_TRIG1	TIOA0 TC0
2	ADC_TRIG2	TIOA1 TC0
3	ADC_TRIG3	TIOA2 TC0
4	ADC_TRIG4	TIOA0_TC1
5	ADC_TRIG5	ACC
6	ADC_TRIG6	PWM event line 0
7	ADC_TRIG7	RTCOUT0

38.6.3 ADC Channel Sequence Register 1

Name: ADC_SEQR1
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USCH8[3:0]				USCH7[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USCH6[3:0]				USCH5[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USCH4[3:0]				USCH3[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USCH2[3:0]				USCH1[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31, 32:35 – USCHx User Sequence Number x
The allowed range is 0 to 15, thus only a CH0 to CH15 sequence can be used.
This register can be used only if ADC_MR.USEQ is set to ‘1’.
Any USCHx field is processed only if the bit CHx-1 in ADC_CHSR reads logical ‘1’, else any value written in USCHx does not add the corresponding channel in the conversion sequence.
Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.
Example: For each trigger event, to obtain the “CH3 CH1 CH0 CH4 CH4” conversion sequence, use the following settings:
ADC_SEQR1.USCH1=3, ADC_CHSR.CH0=1
ADC_SEQR1.USCH2=1, ADC_CHSR.CH1=1
ADC_SEQR1.USCH3=0, ADC_CHSR.CH2=1
ADC_SEQR1.USCH4=4, ADC_CHSR.CH3=1
ADC_SEQR1.USCH5=4, ADC_CHSR.CH4=1

38.6.4 ADC Channel Sequence Register 2

Name: ADC_SEQR2
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USCH16[3:0]				USCH15[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USCH14[3:0]				USCH13[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USCH12[3:0]				USCH11[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USCH10[3:0]				USCH9[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – USCHx User Sequence Number x
The allowed range is 0 to 15, thus only a CH0 to CH15 sequence can be used.
This register can be used only if ADC_MR.USEQ is set to ‘1’.
Any USCHx field is processed only if the CHx-1 bit in ADC_CHSR reads logical ‘1’, else any value written in USCHx does not add the corresponding channel in the conversion sequence.
Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.

38.6.5 ADC Channel Enable Register

Name: ADC_CHER
Offset: 0x10
Reset: –
Property: Write-only

If ADC_MR.USEQ = 1, CHx corresponds to the enable of sequence number x+1 described in ADC_SEQR1 and ADC_SEQR2 (e.g. CH0 enables sequence number USCH1). For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CH31	CH30						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 30, 31 – CH30, CH31 Channel x Enable

Value	Description
0	No effect.
1	Enables the corresponding channel.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CHx Channel x Enable

Value	Description
0	No effect.
1	Enables the corresponding channel.

38.6.6 ADC Channel Disable Register

Name: ADC_CHDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

⚠WARNING If the corresponding channel is disabled during a conversion, or if it is disabled and then reenabled during a conversion, its associated data and corresponding ADC_EOC_ISR.EOCx, ADC_ISR.GOVRE and ADC_OVER.OVREx flags are unpredictable.

Bit	31	30	29	28	27	26	25	24
	CH31	CH30						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 30, 31 – CH30, CH31 Channel x Disable

Value	Description
0	No effect.
1	Disables the corresponding channel.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CHx Channel x Disable

Value	Description
0	No effect.
1	Disables the corresponding channel.

38.6.7 ADC Channel Status Register

Name: ADC_CHSR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CH31	CH30						
Access	R	R						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 30, 31 – CH30, CH31 Channel x Status

Value	Description
0	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is disabled..
1	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CHx Channel x Status
As an example, when ADC_MR.USEQ=1 and ADC_CHSR.CH2=1, the channel configured in ADC_SEQ1R.USCH3 is part of the sequence of conversions.

Value	Description
0	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is disabled..
1	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is enabled.

38.6.8 ADC Last Converted Data Register

Name: ADC_LCDR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				CHNBOSR[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – CHNBOSR[4:0] Channel Number in Oversampling Mode
Indicates the last converted channel when ADC_EMR.TAG is set and ADC_EMR.OSR is not equal to 0. If ADC_EMR.TAG is not set, CHNBOSR = 0.

Bits 15:0 – LDATA[15:0] Last Data Converted
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.
If OSR = 0 and TAG = 1 in ADC_EMR, the 4 MSBs of LDATA carry the channel number to obtain a packed system memory buffer made of 1 converted data stored in a half-word (16 bits) instead of 1 converted data in a 32-bit word, thus dividing by 2 the size of the memory buffer. See ADC_LCDR (NO_OSR).

38.6.9 ADC Last Converted Data Register (NO_OSR)

Name: ADC_LCDR (NO_OSR)
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	NO_OSR_CHNBOSR[3:0]				NO_OSR_LDATA[11:8]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NO_OSR_LDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – NO_OSR_CHNBOSR[3:0] Channel Number when No Oversampling
Indicates the last converted channel when ADC_EMR.TAG is set and the ADC_EMR.OSR = 0. If ADC_EMR.TAG is not set, NO_OSR_CHNB = 0.

Bits 11:0 – NO_OSR_LDATA[11:0] Last Data Converted when No Oversampling
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.

38.6.10 ADC Interrupt Enable Register

Name: ADC_IER
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
						COMPE	GOVRE	DRDY
Access						W	W	W
Reset						–	–	–

Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					W	W		
Reset					–	–		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 26 – COMPE Comparison Event Interrupt Enable

Bit 25 – GOVRE General Overrun Error Interrupt Enable

Bit 24 – DRDY Data Ready Interrupt Enable

Bit 19 – TEMPCHG Temperature Change Interrupt Enable

Bit 18 – EOS End Of Sequence Interrupt Enable

Bit 5 – RXOVR Receive Over Flow Interrupt Enable

Bit 4 – RXUDR Receive Under Flow Interrupt Enable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Enable

Bit 2 – RXFULL Receive FIFO Full Interrupt Enable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Enable

Bit 0 – RXRDY Receive Ready Interrupt Enable

38.6.11 ADC Interrupt Disable Register

Name: ADC_IDR
Offset: 0x28
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
						COMPE	GOVRE	DRDY
Access						W	W	W
Reset						–	–	–
Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					W	W		
Reset					–	–		
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 26 – COMPE Comparison Event Interrupt Disable

Bit 25 – GOVRE General Overrun Error Interrupt Disable

Bit 24 – DRDY Data Ready Interrupt Disable

Bit 19 – TEMPCHG Temperature Change Interrupt Disable

Bit 18 – EOS End Of Sequence Interrupt Disable

Bit 5 – RXOVR Receive Over Flow Interrupt Disable

Bit 4 – RXUDR Receive Under Flow Interrupt Disable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Disable

Bit 2 – RXFULL Receive FIFO Full Interrupt Disable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Disable

Bit 0 – RXRDY Receive Ready Interrupt Disable

38.6.12 ADC Interrupt Mask Register

Name: ADC_IMR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
						COMPE	GOVRE	DRDY
Access						R	R	R
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					R	R		
Reset					0	0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 26 – COMPE Comparison Event Interrupt Mask

Bit 25 – GOVRE General Overrun Error Interrupt Mask

Bit 24 – DRDY Data Ready Interrupt Mask

Bit 19 – TEMPCHG Temperature Change Interrupt Disable

Bit 18 – EOS End Of Sequence Interrupt Mask

Bit 5 – RXOVR Receive Over Flow Interrupt Mask

Bit 4 – RXUDR Receive Under Flow Interrupt Mask

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Mask

Bit 2 – RXFULL Receive FIFO Full Interrupt Mask

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Mask

Bit 0 – RXRDY Receive Ready Interrupt Mask

38.6.13 ADC Interrupt Status Register

Name: ADC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						COMPE	GOVRE	DRDY
Access						R	R	R
Reset						0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
					TEMPCHG	EOS		
Access					R	R		
Reset					0	0		
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 26 – COMPE Comparison Event (cleared on read)

Value	Description
0	No comparison event occurred since the last read of ADC_ISR.
1	At least one comparison event (defined in ADC_EMR and ADC_CWR) has occurred since the last read of ADC_ISR.

Bit 25 – GOVRE General Overrun Error (cleared on read)

Value	Description
0	No general overrun error occurred since the last read of ADC_ISR.
1	At least one general overrun error has occurred since the last read of ADC_ISR.

Bit 24 – DRDY Data Ready (automatically set / cleared)

Value	Description
0	No data has been converted since the last read of ADC_LCDR.
1	At least one data has been converted and is available in ADC_LCDR.

Bit 19 – TEMPCHG Temperature Change (cleared on read)

Value	Description
0	There is no comparison match (defined in the Temperature Compare Window register (ADC_TEMPCWR) since the last read of ADC_ISR.
1	The temperature value reported on ADC_CDRmax (max=highest index) has changed since the last read of ADC_ISR, according to what is defined in ADC_TEMPTMR and ADC_TEMPCWR.

Bit 18 – EOS End Of Sequence (cleared on read)

Value	Description
0	No sequence is in progress or the sequence is not finished. This flag is cleared when reading ADC_ISR.
1	The sequence is complete.

Bit 5 – RXOVR Receive Over Flow (cleared on read)

Value	Description
0	No general overrun error occurred since the last read of ADC_ISR.
1	At least one general overrun error has occurred since the last read of ADC_ISR.

Bit 4 – RXUDR Receive Under Flow (cleared on read)

Value	Description
0	No general underrun error occurred since the last read of ADC_ISR.
1	At least one general underrun error has occurred since the last read of ADC_ISR.

Bit 3 – RXCHUNK Receive FIFO Chunk (cleared on read)

Value	Description
0	The number of written elements in the FIFO has been lower than or not equal to chunk_size since the last read of ADC_ISR.
1	The number of written elements in the FIFO has been greater than or equal to chunk_size since the last read of ADC_ISR.

Bit 2 – RXFULL Receive FIFO Full (cleared on read)

Value	Description
0	FIFO has not been full since the last read of ADC_ISR.
1	FIFO has been full since the last read of ADC_ISR.

Bit 1 – RXEMPTY Receive FIFO Empty (cleared on read)

Value	Description
0	FIFO has not been empty since the last read of ADC_ISR.
1	FIFO has been empty since the last read of ADC_ISR.

Bit 0 – RXRDY Receive Ready (cleared on read)

Value	Description
0	FIFO has been empty since the last read of ADC_ISR.
1	One element has been written since the last read of ADC_ISR.

38.6.14 ADC End Of Conversion Interrupt Enable Register

Name: ADC_EOC_IER
Offset: 0x34
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	EOC31	EOC30						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 30, 31 – EOC30, EOC31 End of Conversion Interrupt Enable x

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EOCx End of Conversion Interrupt Enable x

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

38.6.15 ADC End Of Conversion Interrupt Disable Register

Name: ADC_EOC_IDR
Offset: 0x38
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	EOC31	EOC30						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 30, 31 – EOC30, EOC31 End of Conversion Interrupt Disable x

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EOCx End of Conversion Interrupt Disable x

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

38.6.16 ADC End Of Conversion Interrupt Mask Register

Name: ADC_EOC_IMR
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EOC31	EOC30						
Access	W	W						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 30, 31 – EOC30, EOC31 End of Conversion Interrupt Mask x

Value	Description
0	The corresponding interrupt is disabled.
1	The corresponding interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EOCx End of Conversion Interrupt Mask x

Value	Description
0	The corresponding interrupt is disabled.
1	The corresponding interrupt is enabled.

38.6.17 ADC End Of Conversion Interrupt Status Register

Name: ADC_EOC_ISR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EOC31	EOC30						
Access	W	W						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EOC15	EOC14	EOC13	EOC12	EOC11	EOC10	EOC9	EOC8
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 30, 31 – EOC30, EOC31 End of Conversion Interrupt Enable x

Value	Description
0	The corresponding analog channel is disabled, or the conversion is not finished. This flag is cleared when reading the corresponding ADC_CDRx registers.
1	The corresponding analog channel is enabled and conversion is complete.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EOCx End of Conversion x (automatically set / cleared)

Value	Description
0	The corresponding analog channel is disabled, or the conversion is not finished. This flag is cleared when reading the corresponding ADC_CDRx registers.
1	The corresponding analog channel is enabled and conversion is complete.

38.6.18 ADC Temperature Sensor Mode Register

Name: ADC_TEMPMR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TEMPCMPMOD[1:0]					TEMPON
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 5:4 – TEMPCMPMOD[1:0] Temperature Comparison Mode

Value	Name	Description
0	LOW	Generates the TEMPCHG flag in ADC_ISR when the converted data is lower than the low threshold of the window.
1	HIGH	Generates the TEMPCHG flag in ADC_ISR when the converted data is higher than the high threshold of the window.
2	IN	Generates the TEMPCHG flag in ADC_ISR when the converted data is in the comparison window.
3	OUT	Generates the TEMPCHG flag in ADC_ISR when the converted data is out of the comparison window.

Bit 0 – TEMPON Temperature Sensor On

Value	Description
0	The temperature sensor is not enabled.
1	The temperature sensor is enabled and the measurements are triggered.

38.6.19 ADC Temperature Compare Window Register

Name: ADC_TEMPCWR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					THIGHTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	THIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					TLOWTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TLOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – THIGHTHRES[11:0] Temperature High Threshold
High threshold associated to ADC_TEMPMR compare settings.

Bits 11:0 – TLOWTHRES[11:0] Temperature Low Threshold
Low threshold associated to ADC_TEMPMR compare settings.

38.6.20 ADC Overrun Status Register

Name: ADC_OVER
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	OVRE31	OVRE30						
Access	W	W						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVRE15	OVRE14	OVRE13	OVRE12	OVRE11	OVRE10	OVRE9	OVRE8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 30, 31 – OVRE30, OVRE31 Overrun Error x

Value	Description
0	No overrun error has occurred on the corresponding channel since the last read of ADC_OVER.
1	An overrun error has occurred on the corresponding channel since the last read of ADC_OVER.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – OVREx Overrun Error x
An overrun error does not always mean that the unread data has been replaced by new valid data. See Enhanced Resolution Mode and Digital Averaging Function for details.

Value	Description
0	No overrun error has occurred on the corresponding channel since the last read of ADC_OVER.
1	An overrun error has occurred on the corresponding channel since the last read of ADC_OVER.

38.6.21 ADC Extended Mode Register

Name: ADC_EMR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		ALTCH	ADCMODE[1:0]			SIGNMODE[1:0]		TAG
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	TRACKX[1:0]		ALWAYS0	ASTE		OSR[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
			CMPFILTER[1:0]				CMPALL	CMPSEL[4]
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 30 – ALTCH Alternate Channel Selection (Safety)

Value	Description
0	The regular channels are selected.
1	The alternate channels are selected.

Bits 29:28 – ADCMODE[1:0] ADC Running Mode

See Automatic Error Correction for details on ADC Running mode.

Value	Name	Description
0	NORMAL	Normal mode of operation.
1	OFFSET_ERROR	Offset Error mode to measure the offset error. See the table ADC Running Modes.
2	GAIN_ERROR_HIGH	Gain Error mode to measure the gain error. See the table ADC Running Modes.
3	GAIN_ERROR_LOW	Gain Error mode to measure the gain error. See the table ADC Running Modes.

Bits 26:25 – SIGNMODE[1:0] Sign Mode

If conversion results are signed and resolution is below 16 bits, the sign is extended up to bit 15 (for example, 0xF43 for 12-bit resolution is read as 0xFF43 and 0x467 is read as 0x0467). See Conversion Results Format.

Value	Name	Description
0	SE_UNSG_DF_SIGN	Single-ended channels: unsigned conversions Differential channels: signed conversions
1	SE_SIGN_DF_UNSG	Single-ended channels: signed conversions Differential channels: unsigned conversions
2	ALL_UNSIGNED	All channels: unsigned conversions
3	ALL_SIGNED	All channels: signed conversions

Bit 24 – TAG ADC_LCDR Tag

Value	Description
0	Sets ADC_LCDR.NO_OSR_CHNB/CHNBOSR to zero.
1	Appends the channel number to the conversion result in ADC_LCDR.

Bits 23:22 – TRACKX[1:0] Tracking Time x4, x8 or x16

Value	Name	Description
0	TRACKTIMx1	ADC_MR.TRACKTIM effect is multiplied by 1.
1	TRACKTIMx4	ADC_MR.TRACKTIM effect is multiplied by 4.
2	TRACKTIMx8	ADC_MR.TRACKTIM effect is multiplied by 8.
3	TRACKTIMx16	ADC_MR.TRACKTIM effect is multiplied by 16.

Bit 21 – ALWAYS0 Must always be written to 0

Bit 20 – ASTE Averaging on Single Trigger Event

Value	Name	Description
0	MULTI_TRIG_AVERAGE	The average requests several trigger events.
1	SINGLE_TRIG_AVERAGE	The average requests only one trigger event.

Bits 18:16 – OSR[2:0] Over Sampling Rate

Value	Name	Description
0	NO_AVERAGE	No averaging. ADC sample rate is maximum.
1	OSR4	1-bit enhanced resolution by averaging. ADC sample rate divided by 4.
2	OSR16	2-bit enhanced resolution by averaging. ADC sample rate divided by 16.
3	OSR64	1-bit enhanced resolution by averaging. ADC sample rate divided by 64.
4	OSR256	2-bit enhanced resolution by averaging. ADC sample rate divided by 256.

Bits 13:12 – CMPFILTER[1:0] Compare Event Filtering
Number of consecutive compare events necessary to raise the flag = CMPFILTER+1
When programmed to 0, the flag rises as soon as an event occurs.
See Comparison Window when using the filtering option (CMPFILTER > 0).

Bit 9 – CMPALL Compare All Channels

Value	Description
0	Only the channel indicated in CMPSEL is compared.
1	All channels are compared.

Bits 8:4 – CMPSEL[4:0] Comparison Selected Channel
If CMPALL = 0: CMPSEL indicates which channel has to be compared.
If CMPALL = 1: No effect.

Bit 2 – CMPTYPE Comparison Type

Value	Name	Description
0	FLAG_ONLY	Any conversion is performed and comparison function drives the COMPE flag.
1	START_CONDITION	Comparison conditions must be met to start the storage of all conversions until ADC_CR.CMPRST is set.

Bits 1:0 – CMPMODE[1:0] Comparison Mode

Value	Name	Description
0	LOW	When the converted data is lower than the low threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.
1	HIGH	When the converted data is higher than the high threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.
2	IN	When the converted data is in the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.

Value	Name	Description
3	OUT	When the converted data is out of the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.

38.6.22 ADC Compare Window Register

Name: ADC_CWR
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	HIGHTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LOWTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – HIGHTHRES[15:0] High Threshold
High threshold associated to ADC_EMR compare settings.

Bits 15:0 – LOWTHRES[15:0] Low Threshold
Low threshold associated to ADC_EMR compare settings.

38.6.23 ADC Channel Configuration Register

Name: ADC_CCR
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DIFF15	DIFF14	DIFF13	DIFF12	DIFF11	DIFF10	DIFF9	DIFF8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – DIFFx Differential Inputs for Channel x

Value	Description
0	Corresponding channel is set in Single-ended mode.
1	Corresponding channel is set in Differential mode.

38.6.24 ADC Channel Data Register

Name: ADC_CDRx
Offset: 0x60 + x*0x04 [x=0..31]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Converted Data
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed. ADC_CDRx is only loaded if the corresponding analog channel is enabled.

38.6.25 ADC Analog Control Register

Name: ADC_ACR
Offset: 0xE0
Reset: 0x00000100
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

By default, bits 12 and 13 are set to 1 and 0, respectively, and must not be modified.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SRCLCH
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 16 – SRCLCH Source Last Channel

Value	Name	Description
0	VTEMP	The highest index channel is driven by the output of the temperature sensor.
1	VBG	The highest index channel is driven by the reference voltage of the temperature sensor.

38.6.26 ADC FIFO Mode Register

Name: ADC_FMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FIFOCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHUNK[3:0]						ENLEVEL	ENFIFO
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bits 23:16 – FIFOCNT[7:0] FIFO Count
Number of conversions available in the FIFO (not a source of interrupt).

Bits 7:4 – CHUNK[3:0] Chunk Size
Number of elements in FIFO required to generate a DMA request. Minimum value is 1. Maximum value is 15.

Bit 1 – ENLEVEL Enable Level

Value	Description
0	Request to DMA is generated as soon as one data is written in FIFO when FIFO is enabled. CHUNK is not used.
1	Request to DMA is generated as soon as the number of written elements in the FIFO is greater than or equal to CHUNK.

Bit 0 – ENFIFO Enable FIFO

Value	Description
0	FIFO is disabled.
1	FIFO is enabled.

38.6.27 ADC Trigger Register

Name: ADC_TRGR
Offset: 0x100
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TRGPER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TRGPER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TRGPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						TRGMOD[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – TRGPER[23:0] Trigger Period
Effective only if TRGMOD defines a periodic trigger.
Defines the periodic trigger period, with the following equation:
Trigger Period = (TRGPER + 1) / ADCCLK
The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence depending on the configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx and ADC_TSMR.

Bits 2:0 – TRGMOD[2:0] Trigger Mode

Value	Name	Description
0	NO_TRIGGER	No trigger, only software trigger can start conversions
1	EXT_TRIG_RISE	Rising edge of the selected trigger event, defined in ADC_MR.TRGSEL
2	EXT_TRIG_FALL	Falling edge of the selected trigger event
3	EXT_TRIG_ANY	Any edge of the selected trigger event
4		–
5	PERIOD_TRIG	ADC internal periodic trigger (see TRGPER)
6	CONTINUOUS	Continuous mode, Free Run mode

38.6.28 ADC Correction Select Register

Name: ADC_COSR
Offset: 0x104
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CSEL[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – CSEL[4:0] Channel Correction Select
Selects the channel to be displayed in ADC_CVR.

38.6.29 ADC Correction Values Register

Name: ADC_CVR
Offset: 0x108
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GAINCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OFFSETCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – GAINCORR[15:0] Gain Correction
Gain correction to apply on converted data. Only bits 0 to 13 are relevant (other bits are ignored and read as 0).

Bits 15:0 – OFFSETCORR[15:0] Offset Correction
Offset correction to apply on converted data. The offset is signed (2's complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

38.6.30 ADC Channel Error Correction Register

Name: ADC_CECR
Offset: 0x10C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ECORR31	ECORR30						
Access	R/W	R/W						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ECORR15	ECORR14	ECORR13	ECORR12	ECORR11	ECORR10	ECORR9	ECORR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30, 31 – ECORR30, ECORR31 Error Correction Enable for Channel x

Value	Description
0	Automatic error correction is disabled for channel x.
1	Automatic error correction is enabled for channel x.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – ECORRx Error Correction Enable for Channel x

Value	Description
0	Automatic error correction is disabled for channel x.
1	Automatic error correction is enabled for channel x.

38.6.31 ADC Write Protection Mode Register

Name: ADC_WPMR
Offset: 0x118
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

38.6.32 ADC Write Protection Status Register

Name: ADC_WPSR
Offset: 0x11C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of ADC_WPSR.
1	A write protection violation has occurred since the last read of ADC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

39. Analog Comparator Controller (ACC)

39.1 Description

The Analog Comparator Controller (ACC) configures the analog comparator and generates an interrupt depending on user settings. The analog comparator embeds two 8-to-1 multiplexers that generate two internal inputs. These inputs are compared, resulting in a compare output. The hysteresis level, edge detection and polarity are configurable.

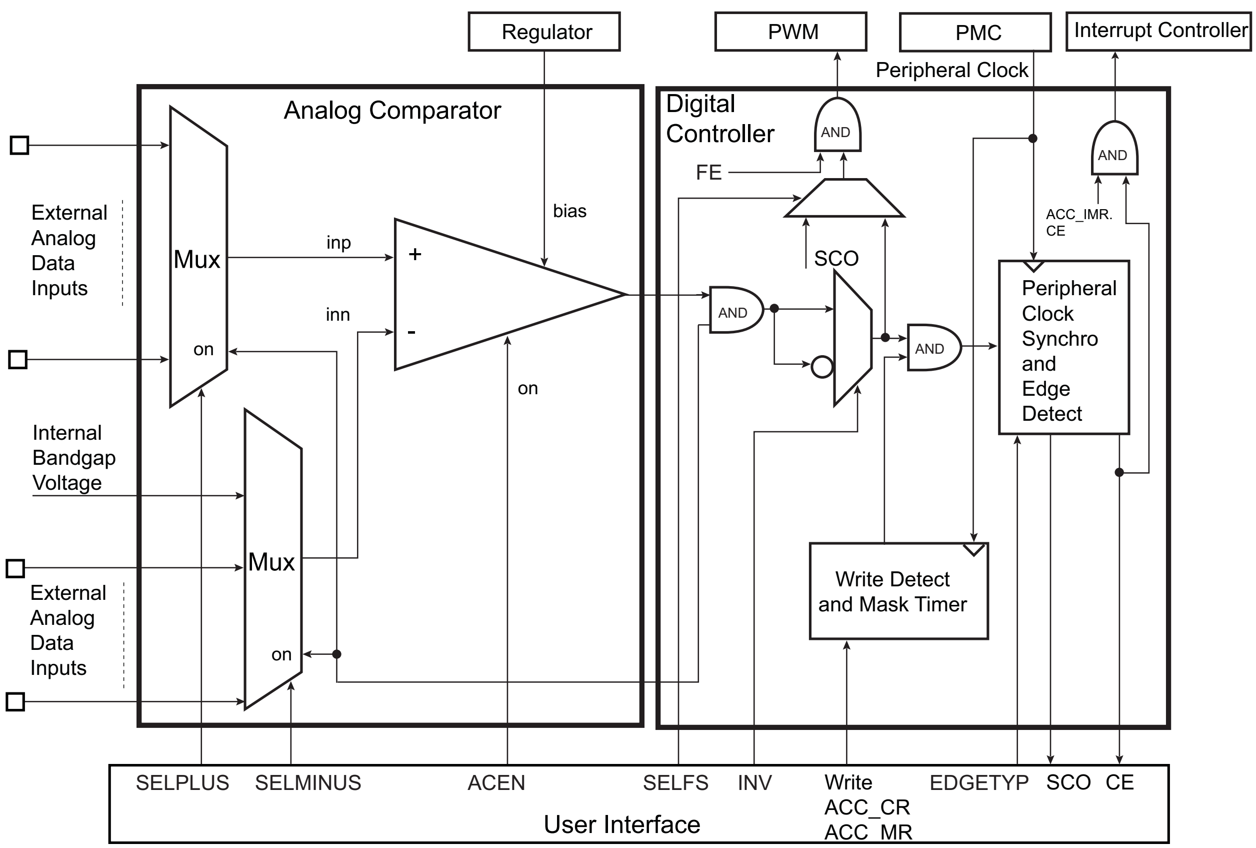
The ACC also generates a compare event which can be used by the Pulse Width Modulator (PWM).

39.2 Embedded Characteristics

- 7 User Analog Inputs Selectable for Comparison
- Bandgap Voltage Reference Selectable for Comparison
- Interrupt Generation
- Compare Event Fault Generation for PWM

39.3 Block Diagram

Figure 39-1. ACC Block Diagram



39.4 Signal Description

Table 39-1. ACC Signal Description

Pin Name	Description	Type
ACC_INP0..3, ACC_INN1..3	External analog data inputs	Input
VBG[(1)]	Internal bandgap voltage	Input

Note:
1. To enable the VBG band gap voltage (refer to the section Analog-to-Digital Converter (ADC) Controller) and enable the temperature sensor VTEMP. This also enables the VBG.

39.5 Product Dependencies

39.5.1 I/O Lines

The analog input pins (ACC_INP0..3 and ACC_INN1..3) are multiplexed with digital functions (PIO) on the IO line. By writing the SELMINUS and SELPLUS fields in the ACC Mode Register (ACC_MR), the associated IO lines are set to Analog mode.

39.5.2 Power Management

The ACC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the ACC clock.

Note that the voltage regulator must be activated to use the analog comparator.

39.5.3 Interrupt Sources

The ACC has an interrupt line connected to the Interrupt Controller (IC). In order to handle interrupts, the Interrupt Controller must be programmed before configuring the ACC.

39.5.4 Fault Output

The ACC has the FAULT output connected to the FAULT input of PWM. See Fault Mode and the implementation of the PWM in the product.

39.6 Functional Description

39.6.1 Description

The Analog Comparator Controller (ACC) controls the analog comparator settings and performs postprocessing of the analog comparator output.

When the analog comparator settings are modified, the output of the analog cell may be invalid. The ACC masks the output for the invalid period.

A comparison flag is triggered by an event on the output of the analog comparator and an interrupt is generated. The event on the analog comparator output can be selected among falling edge, rising edge or any edge.

The ACC registers are listed in the Register Summary.

39.6.2 Fault Mode

In Fault mode, a comparison match event is communicated by the ACC fault output which is directly and internally connected to a PWM fault input.

The source of the fault output can be configured as either a combinational value derived from the analog comparator output or as the peripheral clock resynchronized value. Refer to Block Diagram.

39.6.3 Output Masking Period

As soon as the analog comparator settings change, the output is invalid for a duration. A masking period is automatically triggered as soon as a write access is performed on the ACC_MR and ACC_CR regardless of the register data content.

The mask period is 8 × $t_{peripheral\ clock}$ if ACC_ACR.MSEL = 0.

The mask period is 128 × $t_{peripheral\ clock}$ if ACC_ACR.MSEL = 1.

The masking period is reported by reading a negative value (MASK='1') on the ACC Interrupt Status register (ACC_ISR).

39.6.4 Register Write Protection

To prevent any single software error from corrupting ACC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the ACC Write Protection Mode Register (ACC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the ACC Write Protection Status Register is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the ACC_WPSR register.

The following registers can be write-protected:

- ACC Mode Register
- ACC Analog Control Register

39.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ACC_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	ACC_MR	31:24								
		23:16								
		15:8		FE	SELFS	INV		EDGETYP[1:0]		ACEN
		7:0		SELPLUS[2:0]				SELMINUS[2:0]		
0x08 ... 0x23	Reserved									
0x24	ACC_IER	31:24								
		23:16								
		15:8								
		7:0								CE
0x28	ACC_IDR	31:24								
		23:16								
		15:8								
		7:0								CE
0x2C	ACC_IMR	31:24								
		23:16								
		15:8								
		7:0								CE
0x30	ACC_ISR	31:24	MASK							
		23:16								
		15:8								
		7:0							SCO	CE
0x34 ... 0x93	Reserved									
0x94	ACC_ACR	31:24								
		23:16								
		15:8								
		7:0								MSEL
0x98 ... 0xE3	Reserved									
0xE4	ACC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	ACC_WPSR	31:24								
		23:16								
		15:8								
		7:0								WPVS

39.7.1 ACC Control Register

Name: ACC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the module.

39.7.2 ACC Mode Register

Name: ACC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ACC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		FE	SELFS	INV		EDGETYP[1:0]		ACEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		SELPLUS[2:0]				SELMINUS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 14 – FE Fault Enable

Value	Name	Description
0	DIS	The FAULT output is tied to 0.
1	EN	The FAULT output is driven by the signal defined by SELFS.

Bit 13 – SELFS Selection Of Fault Source

Value	Name	Description
0	CE	The CE flag is used to drive the FAULT output.
1	OUTPUT	The output of the analog comparator flag is used to drive the FAULT output.

Bit 12 – INV Invert Comparator Output

Value	Name	Description
0	DIS	Analog comparator output is directly processed.
1	EN	Analog comparator output is inverted prior to being processed.

Bits 10:9 – EDGETYP[1:0] Edge Type

Value	Name	Description
0	RISING	Only rising edge of comparator output
1	FALLING	Falling edge of comparator output
2	ANY	Any edge of comparator output

Bit 8 – ACEN Analog Comparator Enable

Value	Name	Description
0	DIS	Analog comparator disabled.
1	EN	Analog comparator enabled.

Bits 6:4 – SELPLUS[2:0] Selection For Plus Comparator Input
0..3: Selects the input to apply on analog comparator SELPLUS comparison input.

Value	Name	Description
0	ACC_INP0	Selects ACC_INP0
1	ACC_INP1	Selects ACC_INP1
2	ACC_INP2	Selects ACC_INP2
3	ACC_INP3	Selects ACC_INP3

Bits 2:0 – SELMINUS[2:0] Selection for Minus Comparator Input
0..3: Selects the input to apply on analog comparator SELMINUS comparison input.

Value	Name	Description
0	VBG	Selects VBG
1	ACC_INN1	Selects ACC_INN1
2	ACC_INN2	Selects ACC_INN2
3	ACC_INN3	Selects ACC_INN3

39.7.3 ACC Interrupt Enable Register

Name: ACC_IER
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								W
Reset								–

Bit 0 – CE Comparison Edge

Value	Description
0	No effect.
1	Enables the interrupt when the selected edge (defined by EDGETYP) occurs.

39.7.4 ACC Interrupt Disable Register

Name: ACC_IDR
Offset: 0x28
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								W
Reset								–

Bit 0 – CE Comparison Edge

Value	Description
0	No effect.
1	Disables the interrupt when the selected edge (defined by EDGETYP) occurs.

39.7.5 ACC Interrupt Mask Register

Name: ACC_IMR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								R
Reset								0

Bit 0 – CE Comparison Edge

Value	Description
0	The interrupt is disabled.
1	The interrupt is enabled.

39.7.6 ACC Interrupt Status Register

Name: ACC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	MASK							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SCO	CE
Access							R	R
Reset							0	0

Bit 31 – MASK Flag Mask

Value	Description
0	The CE flag and SCO value are valid.
1	The CE flag and SCO value are invalid.

Bit 1 – SCO Synchronized Comparator Output
Returns an image of the analog comparator output after being preprocessed (refer to Block Diagram).
If INV = 0

- SCO = 0 if inn > inp
- SCO = 1 if inp > inn

If INV = 1

- SCO = 1 if inn > inp
- SCO = 0 if inp > inn

Bit 0 – CE Comparison Edge (cleared on read)

Value	Description
0	No edge occurred (defined by EDGETYP) on analog comparator output since the last read of ACC_ISR.
1	A selected edge (defined by EDGETYP) on analog comparator output occurred since the last read of ACC_ISR.

39.7.7 ACC Analog Control Register

Name: ACC_ACR
Offset: 0x94
Reset: 0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ACC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								MSEL
Access								R/W
Reset								0

Bit 0 – MSEL Masking Period Selection

Value	Description
0	Masks AC output for 16 peripheral clock periods after any write access in ACC_MR or ACC_CR.
1	Masks AC output for 128 peripheral clock periods after any write access in ACC_MR or ACC_CR.

39.7.8 ACC Write Protection Mode Register

Name: ACC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Refer to Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414343	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x414343 (“ACC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x414343 (“ACC” in ASCII).

39.7.9 ACC Write Protection Status Register

Name: ACC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

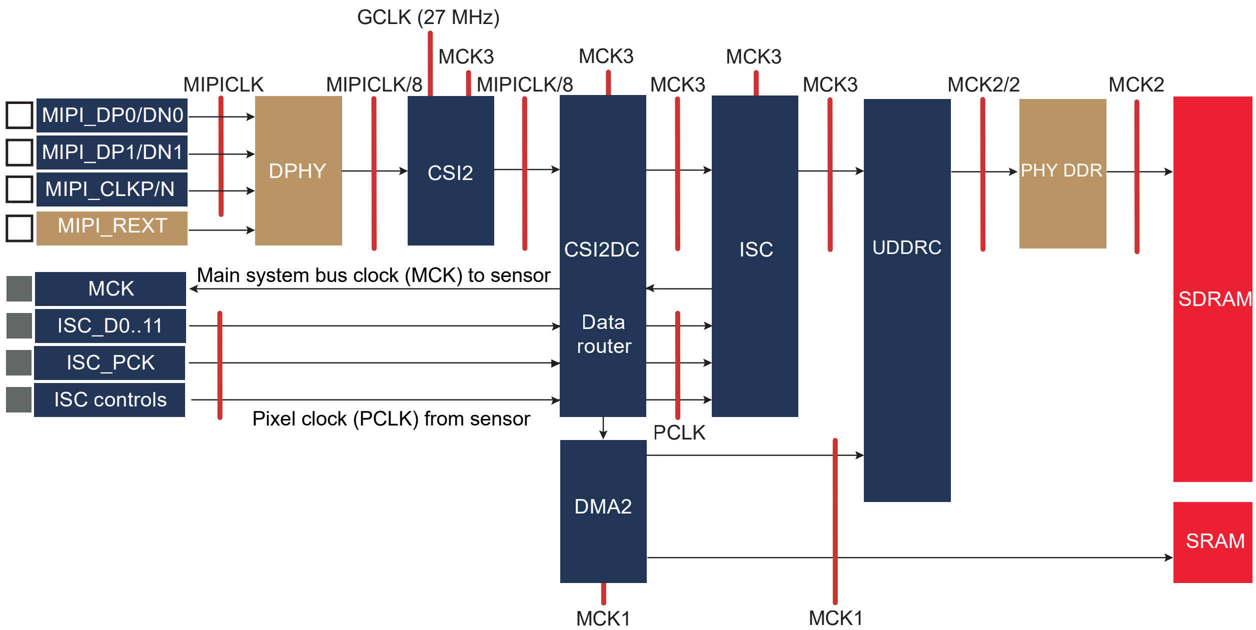
Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last command ACC_CR.SWRST=1.
1	A write protection violation (WPEN = 1) has occurred since the last command ACC_CR.SWRST=1.

40. IMAGE SUBSYSTEM

40.1 Block Diagram

Figure 40-1. Image Subsystem Block Diagram



This block diagram shows peripheral clocks used and clock frequencies on data paths.

40.2 Components

The image subsystem connects MIPI and RGB image sensors to the device and is built with:

- Image Sensor Controller (ISC)
- CSI-2 Demultiplexer Controller (CSI2DC)
- Camera Sensor Host Interface (CSI2)
- Physical interface to connect CSI-2 camera sensors (DPHY)
- Central DMA2, to store non-video metadata in memory

40.3 Product Dependencies

40.3.1 Clocks

In addition to MCK0, MCK1 and MCK2, the following clocks must be started to run the image subsystem:

- CSI2 GCLK input
- MCK3, that can be sourced from a dedicated IMGPLL

All clocks shown in the block diagram are subdivisions of these clocks.

40.3.2 Interrupts

Refer to the table Peripheral Identifiers.

40.3.3 Reset

Image peripherals are connected to the processor and peripherals reset line.

40.3.4 I/Os

MIPI I/Os are high-speed I/Os powered by VDDDPHY/GNDDPHY.

ISC I/Os are multiplexed with other peripherals and powered by VDDIOP0 or VDDIOP1.

40.3.5 Power Saving

To save power consumption, the sensor can stop the image stream. In this case, proceed as follows to enter the image subsystem in Low-power mode:

1. Disable capture in ISC:
   – ISC_CTRLDIS.DISABLE = 1 (to end capture at next vertical synchronization detection)
   – Check ISC_CTRLSR.CAPTURE = 0
2. Power down DPHY:
   – CSI –> CSI2_RESETN = 0
   – CSI –> CSI2_PHY_SHUTDOWNZ = 0
   – CSI –> CSI2_DPHY_RSTZ = 0
3. Reset CSI2DC software:
   – CSI2DC –> CSI2DC_GCTLR = 1

The image subsystem is now in Low-power mode. Proceed as follows to restart the capture:

1. Enable capture in ISC:
   – With color profile update:
     • Write all required configuration registers.
     • ISC_CTRLEN = 0x3 (CAPTURE =1 and UPPRO =1)
   – Without color profile update:
     • ISC_CTRLDIS.CAPTURE = 1
2. Release CSI2DC software reset:
   – CSI2DC –> CSI2DC_GCTLR = 0
3. Enable image pipe:
   – CSI2DC –> CSI2DC_VPER = 1
   – CSI2DC –> CSI2DC_DPER = 1
   – CSI2DC –> CSI2DC_PUR = 0x3
4. Bring DPHY out of power down:
   – CSI –> CSI2_PHY_SHUTDOWNZ = 1
   – CSI –> CSI2_DPHY_RSTZ = 1
   – CSI –> CSI2_RESETN = 1
   – Wait for LP-11 on the clock lane and activate data lanes by probing the CSI2_PHY_STOPSTATE register.

The image datapath can now start streaming frames.

40.4 Special Functions in SFR/SFRBU

None.

41. Camera Serial Interface (CSI)

41.1 Description

The Camera Serial Interface (CSI) receives data from a CSI-2-compliant camera sensor. A D-PHY configured as a client (RX) acts as the physical layer.

The CSI implements the MIPI CSI-2 protocol specification. The CSI-2 link protocol specification is a part of communication protocols defined by MIPI Alliance standards intended for mobile system chip-to-chip communications. The CSI-2 specification is for the image application processor communication in cameras.

41.2 Embedded Characteristics

The CSI is compliant with the following standards:

- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2), Version 1.2
- MIPI Alliance Specification for D-PHY, Version 1.2

D-PHY Characteristics:

- Two-lane Receive MIPI D-PHY Compliant with Revision 1.1 Specification
- Lane Operation Ranging from 80 Mbps to 1.5 Gbps in Forward Direction
- Aggregate Throughput up to 3 Gbps with Two Data Lanes
- Low-Power Escape Modes and Ultra-Low-Power State

CSI-2 Host Characteristics:

- Dynamically Configurable Multi-Lane Merging
- Long and Short Packet Decoding
- Timing Accurate Signaling of Frame and Line Synchronization Packets
- Frame Formats:
  - General frame or digital interlaced video with or without accurate sync timing
  - Data type (packet or frame level) and virtual channel interleaving
- All Primary and Secondary Data Formats.
  - YUV, RGB, RAW (for details, refer to the section Image Sensor Interface (ISC))
  - Generic 8-bit long packet data types
  - User-defined byte-based data
- Error Detection and Correction
  - PHY level
  - Packet level
  - Line level
  - Frame level
- EMI Mitigation

41.3 Signal Description

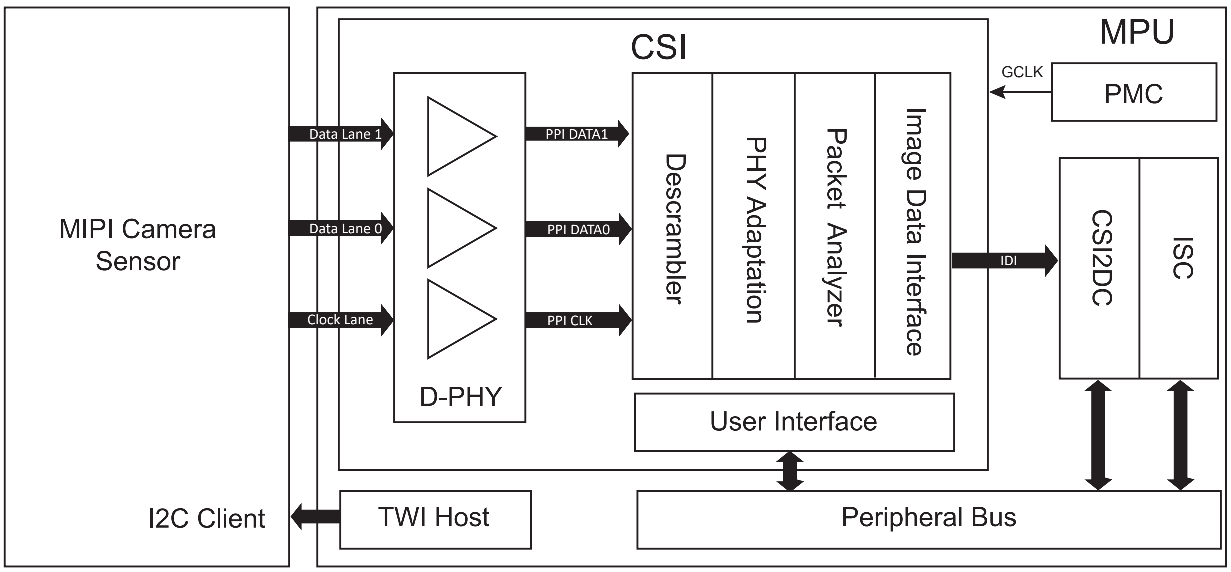
Signal Name	Type	Function
MIPI_DP0	Input	Positive D-PHY differential data line receiver, Lane 0
MIPI_DN0	Input	Negative D-PHY differential data line receiver, Lane 0
MIPI_DP1	Input	Positive D-PHY differential data line receiver, Lane 1

...........continued

Signal Name	Type	Function
MIPI_DN1	Input	Negative D-PHY differential data line receiver, Lane 1
MIPI_CLKP	Input	Positive D-PHY differential clock line receiver
MIPI_CLKN	Input	Negative D-PHY differential clock line receiver
MIPI_REXT	Input	D-PHY external resistor connection

41.4 Block Diagram

Figure 41-1. CSI Block Diagram



CSI main blocks:

- Descrambler: Converts the PHY-Protocol Interface (PPI) scrambled data to its original values.
- PHY Adaptation Layer: Manages the PHY interface, including PHY error handling.
- Packet Analyzer: Data lane merging, together with header decoding, error detection and correction, frame size error detection, and CRC error detection.
- Image Data Interface (IDI): Separates CSI-2 packet header information and reorders the data according to memory storage format. Also generates the timing accurate video synchronization signals. Several error detections are also performed at frame level and line level.
- User Interface: Registers for configuration and control, fully programmable interrupt generator, notifies and monitors the error conditions in the CSI.

Note: Refer to the sections “Image Sensor Controller (ISC)” and “CSI-2 Demultiplexer Controller (CSI2DC)” for details.

41.5 Product Dependencies

41.5.1 Power Management

The CSI is not continuously clocked. Before using it, the programmer must first enable the CSI peripheral and generic (GCLK) clocks in the Power Management Controller.

41.5.2 Interrupt Sources

The CSI interrupt line is connected to one of the internal sources of the interrupt controller. Using the CSI interrupt requires prior programming of the interrupt controller.

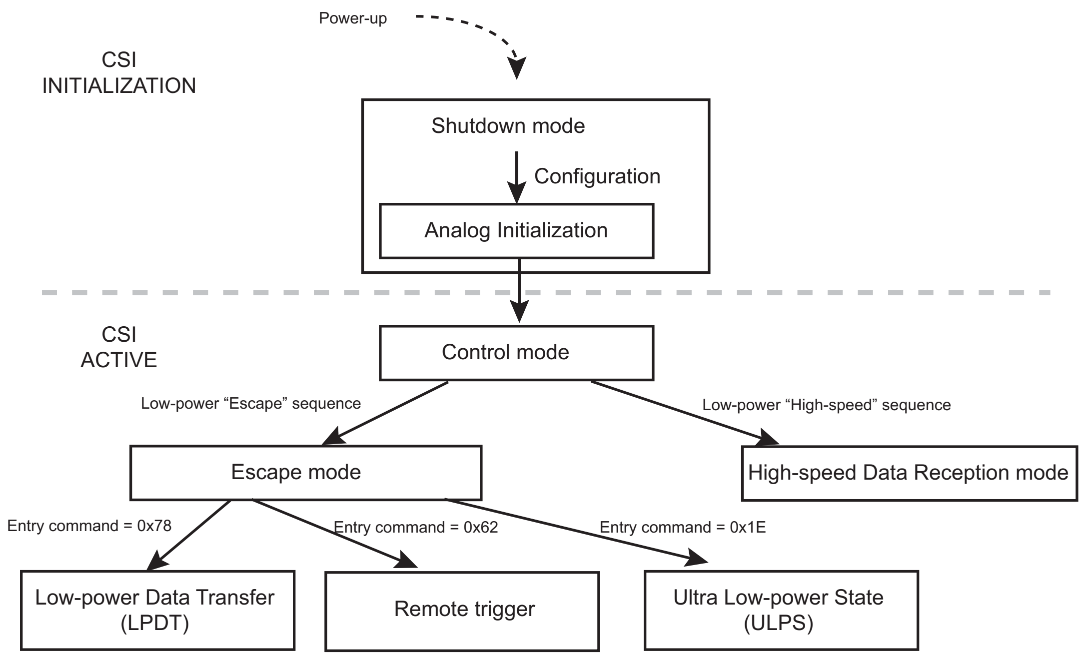
41.6 Functional Description

The CSI interfaces the Image Sensor Controller (ISC) and a MIPI CSI-2 compliant Camera Sensor.

41.6.1 D-PHY Operating Modes

The following figure illustrates the D-PHY initialization from Shutdown to Active modes.

Figure 41-2. D-PHY Initialization from Shutdown to Active Modes



41.6.1.1 Power-Up Mode

The power-up sequence is as follows:

1. Core voltage (VDDCORE) power-up
2. I/O voltage (VDDPHY) power-up

41.6.1.2 Shutdown Mode

Shutdown mode is the operating mode with the lowest power consumption. All analog blocks are disabled, and digital logic is reset. To enter this mode, write to ‘0’ the bits DPHY_RSTZ in the PHY Reset register (CSI_DPHY_RSTZ) and DPHY_SHUTDOWNZ in the PHY Shutdown register (CSI_DPHY_SHUTDOWNZ).

In Shutdown mode, the differential lines of MIPI_DNx/Px and MIPI_CLKN/P are high impedance (Hi-Z).

By default, the D-PHY is configured to work only on the lower operating range of 80-110 Mbps. If higher bit rate operation is required, the register hsfreqrange (HS RX Control of Lane 0) must be set with the proper code. If the D-PHY is expected to change the bit rate after initialization, hsfreqrange should be updated while in Control mode.

When CSI_DPHY_RSTZ.DPHY_RSTZ and CSI_DPHY_SHUTDOWNZ.DPHY_SHUTDOWNZ are set to ‘1’, the D-PHY exits Shutdown mode and starts an initialization procedure.

41.6.1.3 Analog Initialization

The D-PHY comprises 2 data lanes, but some applications may require only one. The number of lanes is configured in the Lane Configuration register (CSI_N_LANES) and must be done only when the D-PHY is in Shutdown mode.

Before starting normal operation, a D-PHY bit rate code must be configured as shown in the following table. Follow the steps listed below for configuration:

1. Ensure that the D-PHY is in Shutdown mode. See Shutdown Mode.
2. Reset the analog configuration by generating a high pulse on CSI_DPHY_TEST_CTRL0. DPHY_TESTCLR.
3. Write a ‘1’ to CSI_DPHY_TEST_CTRL0.DPHY_TESTCLK.
4. Write 0x44 to CSI_DPHY_TEST_CTRL1.DPHY_TESTDIN and write a ‘1’ to CSI_DPHY_TEST_CTRL1.DPHY_TESTEN.
5. Write a ‘0’ to CSI_DPHY_TEST_CTRL0.DPHY_TESTCLK to create a falling edge on DPHY_TESTCLK.
6. Write a ‘0’ to CSI_DPHY_TEST_CTRL1.DPHY_TESTEN and write the configuration value from the following table to CSI_DPHY_TEST_CTRL1.DPHY_TESTDIN.
7. Write a high pulse to CSI_DPHY_TEST_CTRL0.DPHY_TESTCLK by writing ‘1’ immediately followed by ‘0’.

Table 41-1. Bit Rate Ranges

Range (Mbps)	High-Speed Bit Rate Code
80-89	000000
90-99	010000
100-109	100000
110-129	000001
130-139	010001
140-149	100001
150-169	000010
170-179	010010
180-199	100010
200-219	000011
220-239	010011
240-249	100011
250-269	000100
270-299	010100
300-329	000101
330-359	010101
360-399	100101
400-449	000110
450-499	010110
500-549	000111
550-599	010111
600-649	001000
650-699	011000
700-749	001001
750-799	011001
800-849	101001
850-899	111001
900-949	001010

...........continued

Range (Mbps)	High-Speed Bit Rate Code
950-999	011010
1000-1049	101010
1050-1099	111010
1100-1149	001011
1150-1199	011011
1200-1249	101011
1250-1299	111011
1300-1349	001100
1350-1399	011100
1400-1449	101100
1450-1500	111100

The initialization period is a protocol-dependent parameter with a minimum of 100 µs defined by the specification. The D-PHY starts decoding the low-power commands after the analog initialization.

41.6.1.4 Active Modes

41.6.1.4.1 Control Mode

Control mode is the default operating mode. After initialization is completed (analog calibrations), the D-PHY remains in this default mode until a request is placed. The request is placed directly through the sequence of low-power signals in the lanes. The receiver remains in Control mode while receiving Low-power (LP) Stop state (LP-11) in the lines (see the following figure).

Following a request, a lane can exit Control mode for High-speed Data Transfer mode or Escape mode.

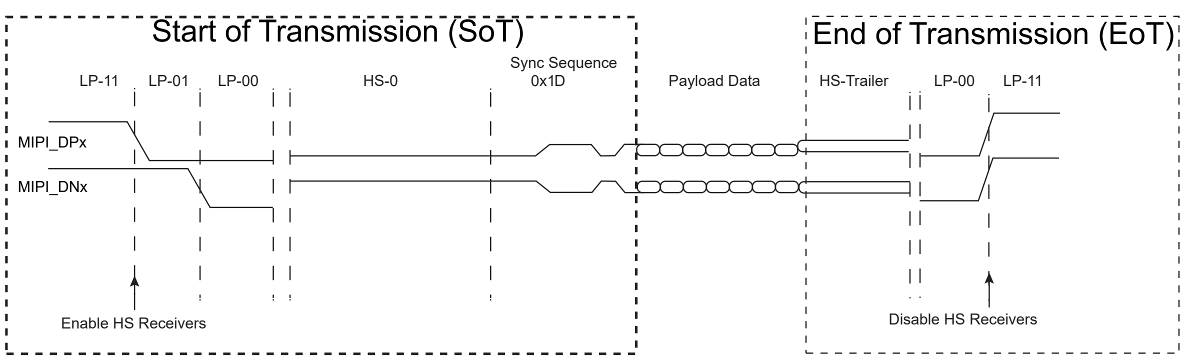
41.6.1.4.2 High-Speed Data Reception Mode

High-speed data reception occurs in bursts. Only during these bursts is the lane in High-Speed mode. A high-speed burst must start from and return to a Stop state (Control mode).

Each data lane can receive a high-speed transmission independently of the state of the remaining data lanes.

A burst contains the low-power initialization sequence, the high-speed data payload, and the end-of-transmission sequence.

Figure 41-3. HS Data Reception Sequence



The D-PHY receiver enters High-Speed mode following the sequence of low-power states in the lines LP-11, LP-01, and LP-00. This sequence is seen as a High-Speed mode request, and toggles the enabling of the high-speed receivers. Synchronization is then achieved through the identification of the leader sequence in the received differential high-speed data. Once the synchronization is achieved, the D-PHY outputs the received bytes through the protocol layer, until a Stop state (LP-11) is detected in the lane.

41.6.1.4.3 Escape Mode

Escape mode is a special mode of operation that uses the data lanes to communicate asynchronously using the low-power states at low speed. A data lane enters Escape mode through an Escape mode entry procedure (LP-11, LP-10, LP-00, LP-01, LP-00). If an LP-11 is detected before reaching LP-00 state, the entry is aborted and the receiver returns to the Stop state. Once the sequence is correctly completed, the transmitter sends an 8-bit command to indicate a requested action. The table below shows the actions supported by Escape mode. If the entry command is invalid, the MAIN_DPHY_ERRESC_x error flag goes high, and the receiver waits until the transmitter returns to the Stop state.

Table 41-2. Possible Escape Mode Sequences for Data Lanes

Escape Mode Action	Entry Command Pattern (First Bit to Last Bit to be Transmitted)	Command Type
Ultra-Low-Power State	0x1E	mode
Reset Trigger	0x62	trigger
Low-Power Data Transmission	0x87	mode

Low-Power Data Transmission (LPDT)

In LPDT mode, the data can be received on the lines at low speed in Low-Power mode. High-speed receivers are off and low-power receivers are on. During LPDT, the protocol can pause by maintaining a Space state on the lines.

Remote Trigger

Remote Trigger mode allows the protocol to send a flag to the receiving side at the request of the transmitting side.

In Escape mode, the RXCLKESC can stop at anytime in either Low or High state.

Ultra-Low-Power State (ULPS)

Ultra-Low-Power State (ULPS) mode has the lowest power consumption, excluding the Shutdown mode.

For data lanes, this mode is entered by sending an ULPS entry command, after the Escape mode entry command. During this mode, the lines are in the Space state (LP-00). Although the clock lane does not support regular Escape mode, the clock lane supports ULPS.

41.6.2 Supported Resolutions and Frame Rates

The table below presents some predefined and supported camera settings, assuming the following:

- D-PHY: clock lane frequency is in the range of 250 MHz to 750 MHz, which results in a bandwidth of 500 Mbps to 1.5 Gbps for each data lane.
- No significant control/reserved traffic is present on the link when pixel data is being transmitted.

Table 41-3. Examples of Supported Resolutions and Frame Rates

	Number of Pixels with Overhead	Refresh Rate (Hz)	Color Depth (bpp)	CSI-2 Bandwidth (Mbits)	D-PHY at 500 Mbps Number of Lanes	D-PHY at 1 Gbps Number of Lanes	D-PHY at 1.5 Gbps Number of Lanes
Camera Formats							
2 Mpixels	2560000	15	24	922	2	1	1
3 Mpixels	3840000	15	24	1382	–	2	1
5 Mpixels	6400000	15	24	2304	–	–	2
8 Mpixels	10240000	10	24	2458	–	–	2
Video Formats							

...........continued

	Number of Pixels with Overhead	Refresh Rate (Hz)	Color Depth (bpp)	CSI-2 Bandwidth (Mbits)	D-PHY at 500 Mbps Number of Lanes	D-PHY at 1 Gbps Number of Lanes	D-PHY at 1.5 Gbps Number of Lanes
1280x720 pixels (720p)	921600	30	24	664	2	1	1
1280x720 pixels (720p)	921600	60	24	1327	3	2	1

Note: The CSI does not perform data decompression. For example, if a camera transmits the compressed data according to the Annex E of the CSI-2 Specification (data compression for raw data types), the CSI decodes the data as normal non-compressed raw data. The decompression is performed by the CSI2DC block.

41.6.3 Descrambler

Data scrambling is used to mitigate the effects of EMI and RF self-interference. The data that is being transmitted is scrambled with a Pseudo-Random Binary Sequence.

The pseudo-random binary sequence can be different on each lane by configuring the Lane 0/1 Scrambling Seed registers (CSI_SCRAMBLING_SEED1/2). The default seed values after reset of the controller are given in the table below.

Lane	Initial Seed Value
0	0x1008
1	0x1188

The descrambler is enabled using the bit SCRAMBLE_ENABLE in the Descrambler Configuration register (CSI_SCRAMBLING).

Selecting the seed used by the descrambler block is done in the registers CSI_SCRAMBLING_SEED1 and CSI_SCRAMBLING_SEED2.

41.6.4 Interrupts

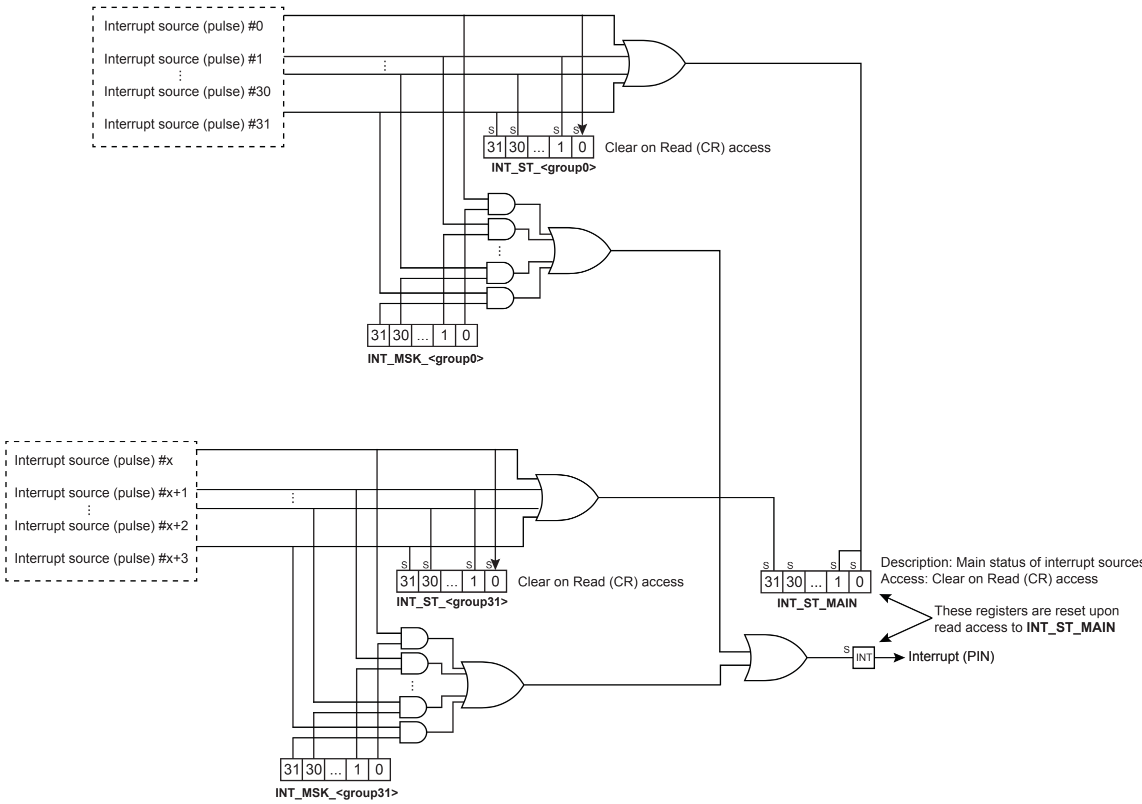
The Interrupt Status registers report error conditions.

The triggering of the CSI interrupt line can be masked by programming the Interrupt Mask registers. By default, all errors are masked. When any bit of these registers is set to 1, it enables the interrupt for a specific error.

The Interrupt Status registers are cleared on read.

The figure below shows the main parts of the interrupt mechanism.

Figure 41-4. Interrupt Mechanism



41.6.5 Error Detection

The CSI analyzes the received packets and determines if there are protocol errors. The following errors are monitored:

- Frame errors, such as incorrect frame sequence, reception of a CRC error in the most recent frame, and the mismatch between Frame Start and Frame End
- Line errors, such as incorrect line sequence and mismatch between Line Start and Line End
- Packet errors, such as payload CRC and ECC (D-PHY)
- D-PHY errors, such as synchronization pattern mismatch

41.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	CSI_N_LANES	31:24								
		23:16								
		15:8								
		7:0						N_LANES[2:0]		
0x08	CSI_CSI2_RESETN	31:24								
		23:16								
		15:8								
		7:0								CSI2_RESETN
0x0C	CSI_INT_ST_MAIN	31:24								
		23:16						STATUS_INT_LINE	STATUS_INT_PKT	STATUS_INT_DPHY
		15:8								
		7:0						STATUS_INT_FRAME_FATAL	STATUS_INT_PKT_FATAL	STATUS_INT_DPHY_FATAL
0x10	CSI_DATA_IDS_1	31:24	DI3_VC[1:0]		DI3_DT[5:0]					
		23:16	DI2_VC[1:0]		DI2_DT[5:0]					
		15:8	DI1_VC[1:0]		DI1_DT[5:0]					
		7:0	DI0_VC[1:0]		DI0_DT[5:0]					
0x14	CSI_DATA_IDS_2	31:24	DI7_VC[1:0]		DI7_DT[5:0]					
		23:16	DI6_VC[1:0]		DI6_DT[5:0]					
		15:8	DI5_VC[1:0]		DI5_DT[5:0]					
		7:0	DI4_VC[1:0]		DI4_DT[5:0]					
0x18 ... 0x3F	Reserved									
0x40	CSI_DPHY_SHUTDOWNZ	31:24								
		23:16								
		15:8								
		7:0								DPHY_SHUTDOWNZ
0x44	CSI_DPHY_RSTZ	31:24								
		23:16								
		15:8								
		7:0								DPHY_RSTZ
0x48	CSI_DPHY_RX	31:24								
		23:16							DPHY_RXCLKACTIVEHS	DPHY_RXULPSCLKNOT
		15:8								
		7:0							DPHY_RXULPSESC_1	DPHY_RXULPSESC_0
0x4C	CSI_DPHY_STOPSTATE	31:24								
		23:16								DPHY_STOPSTATECLK
		15:8								
		7:0							DPHY_STOPSTATEDATA_1	DPHY_STOPSTATEDATA_0
0x50	CSI_DPHY_TEST_CTRL0	31:24								
		23:16								
		15:8								
		7:0							DPHY_TESTCLK	DPHY_TESTCLR

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x54	CSI_DPHY_TEST_CTRL1	31:24								
		23:16								DPHY_TESTEN
		15:8								
		7:0	DPHY_TESTDIN[7:0]							
0x58 ... 0xCB	Reserved									
0xCC	CSI_DPHY_CAL	31:24								
		23:16								
		15:8								
		7:0								RXSKEWCALHS
0xD0 ... 0xDF	Reserved									
0xE0	CSI_INT_ST_DPHY_FATAL	31:24								
		23:16								
		15:8								
		7:0							DPHY_ERRSOTSYNCHS_1	DPHY_ERRSOTSYNCHS_0
0xE4	CSI_INT_MSK_DPHY_FATAL	31:24								
		23:16								
		15:8								
		7:0							MASK_DPHY_ERRSOTSYNCHS_1	MASK_DPHY_ERRSOTSYNCHS_0
0xE8 ... 0xEF	Reserved									
0xF0	CSI_INT_ST_PKT_FATAL	31:24								
		23:16								ERR_ECC_DOUBLE
		15:8								
		7:0					VC3_ERR_CRC	VC2_ERR_CRC	VC1_ERR_CRC	VC0_ERR_CRC
0xF4	CSI_INT_MSK_PKT_FATAL	31:24								
		23:16								MASK_ERR_ECC_DOUBLE
		15:8								
		7:0					MASK_VC3_ERR_CRC	MASK_VC2_ERR_CRC	MASK_VC1_ERR_CRC	MASK_VC0_ERR_CRC
0xF8 ... 0xFF	Reserved									
0x0100	CSI_INT_ST_FRAME_FATAL	31:24								
		23:16					ERR_FRAME_DATA_VC3	ERR_FRAME_DATA_VC2	ERR_FRAME_DATA_VC1	ERR_FRAME_DATA_VC0
		15:8					ERR_F_SEQ_VC3	ERR_F_SEQ_VC2	ERR_F_SEQ_VC1	ERR_F_SEQ_VC0
		7:0					ERR_F_BNDRY_MATCH_VC3	ERR_F_BNDRY_MATCH_VC2	ERR_F_BNDRY_MATCH_VC1	ERR_F_BNDRY_MATCH_VC0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0104	CSI_INT_MSK_FRAME_FATAL	31:24								
		23:16					MASK_ERR_FRAME_DATA_VC3	MASK_ERR_FRAME_DATA_VC2	MASK_ERR_FRAME_DATA_VC1	MASK_ERR_FRAME_DATA_VC0
		15:8					MASK_ERR_F_SEQ_VC3	MASK_ERR_F_SEQ_VC2	MASK_ERR_F_SEQ_VC1	MASK_ERR_F_SEQ_VC0
		7:0					MASK_ERR_F_BNDRY_MATCH_VC3	MASK_ERR_F_BNDRY_MATCH_VC2	MASK_ERR_F_BNDRY_MATCH_VC1	MASK_ERR_F_BNDRY_MATCH_VC0
0x0108 ... 0x010F	Reserved									
0x0110	CSI_INT_ST_DPHY	31:24								
		23:16							DPHY_ERRESC_1	DPHY_ERRESC_0
		15:8								
		7:0							DPHY_ERRSOTHS_1	DPHY_ERRSOTHS_0
0x0114	CSI_INT_MSK_DPHY	31:24								
		23:16							MASK_DPHY_ERRESC_1	MASK_DPHY_ERRESC_0
		15:8								
		7:0							MASK_DPHY_ERRSOTHS_1	MASK_DPHY_ERRSOTHS_0
0x0118 ... 0x011F	Reserved									
0x0120	CSI_INT_ST_PKT	31:24								
		23:16					VC3_ERR_ECC_CORRECTED	VC2_ERR_ECC_CORRECTED	VC1_ERR_ECC_CORRECTED	VC0_ERR_ECC_CORRECTED
		15:8								
		7:0					ERR_ID_VC3	ERR_ID_VC2	ERR_ID_VC1	ERR_ID_VC0
0x0124	CSI_INT_MSK_PKT	31:24								
		23:16					MASK_VC3_ERR_ECC_CORRECTED	MASK_VC2_ERR_ECC_CORRECTED	MASK_VC1_ERR_ECC_CORRECTED	MASK_VC0_ERR_ECC_CORRECTED
		15:8								
		7:0					MASK_ERR_ID_VC3	MASK_ERR_ID_VC2	MASK_ERR_ID_VC1	MASK_ERR_ID_VC0
0x0128 ... 0x012F	Reserved									
0x0130	CSI_INT_ST_LINE	31:24								
		23:16	ERR_L_SEQ_DI7	ERR_L_SEQ_DI6	ERR_L_SEQ_DI5	ERR_L_SEQ_DI4	ERR_L_SEQ_DI3	ERR_L_SEQ_DI2	ERR_L_SEQ_DI1	ERR_L_SEQ_DI0
		15:8								
		7:0	ERR_L_BNDRY_MATCH_DI7	ERR_L_BNDRY_MATCH_DI6	ERR_L_BNDRY_MATCH_DI5	ERR_L_BNDRY_MATCH_DI4	ERR_L_BNDRY_MATCH_DI3	ERR_L_BNDRY_MATCH_DI2	ERR_L_BNDRY_MATCH_DI1	ERR_L_BNDRY_MATCH_DI0
0x0134	CSI_INT_MSK_LINE	31:24								
		23:16	MASK_ERR_L_SEQ_DI7	MASK_ERR_L_SEQ_DI6	MASK_ERR_L_SEQ_DI5	MASK_ERR_L_SEQ_DI4	MASK_ERR_L_SEQ_DI3	MASK_ERR_L_SEQ_DI2	MASK_ERR_L_SEQ_DI1	MASK_ERR_L_SEQ_DI0
		15:8								
		7:0	MASK_ERR_L_BNDRY_MATCH_DI7	MASK_ERR_L_BNDRY_MATCH_DI6	MASK_ERR_L_BNDRY_MATCH_DI5	MASK_ERR_L_BNDRY_MATCH_DI4	MASK_ERR_L_BNDRY_MATCH_DI3	MASK_ERR_L_BNDRY_MATCH_DI2	MASK_ERR_L_BNDRY_MATCH_DI1	MASK_ERR_L_BNDRY_MATCH_DI0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0138 ... 0x02FF	Reserved									
0x0300	CSI_SCRAMBLING	31:24								
		23:16								
		15:8								
		7:0								SCRAMBLE_ENABLE
0x0304	CSI_SCRAMBLING_SEED0	31:24								
		23:16								
		15:8	SCRAMBLE_SEED_LANE0[15:8]							
		7:0	SCRAMBLE_SEED_LANE0[7:0]							
0x0308	CSI_SCRAMBLING_SEED1	31:24								
		23:16								
		15:8	SCRAMBLE_SEED_LANE1[15:8]							
		7:0	SCRAMBLE_SEED_LANE1[7:0]							

41.7.1 CSI Lane Configuration Register

Name: CSI_N_LANES
Offset: 0x4
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						N_LANES[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	1

Bits 2:0 – N_LANES[2:0] Number of active data lanes
The update can be performed only when the D-PHY is in Stop state.

Value	Description
000	One data lane
001	Two data lanes
010	Reserved
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

41.7.2 CSI Reset Control Register

Name: CSI_CSI2_RESETN
Offset: 0x8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CSI2_RESETN
Access								R/W
Reset								0

Bit 0 – CSI2_RESETN Internal controller logic reset

Value	Description
0	Resets the internal controller logic.
1	CSI exits Reset state.

41.7.3 CSI Main Interrupt Status Register

Name: CSI_INT_ST_MAIN
Offset: 0xC
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No event occurred since the last read of the register.

1: An event occurred since the last read of the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						STATUS_INT_LINE	STATUS_INT_PKT	STATUS_INT_DPHY
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						STATUS_INT_FRAME_FATAL	STATUS_INT_PKT_FATAL	STATUS_INT_DPHY_FATAL
Access						R	R	R
Reset						0	0	0

Bit 18 – STATUS_INT_LINE
Indicates if an event occurred in the CSI_INT_ST_LINE register (cleared on read)

Bit 17 – STATUS_INT_PKT
Indicates if an event occurred in the CSI_INT_ST_PKT register (cleared on read)

Bit 16 – STATUS_INT_DPHY
Indicates if an event occurred in the CSI_INT_ST_DPHY register (cleared on read)

Bit 2 – STATUS_INT_FRAME_FATAL
Indicates if an event occurred in the CSI_INT_ST_FRAME_FATAL register (cleared on read)

Bit 1 – STATUS_INT_PKT_FATAL
Indicates if an event occurred in the CSI_INT_ST_PKT_FATAL register (cleared on read)

Bit 0 – STATUS_INT_DPHY_FATAL
Indicates if an event occurred in the CSI_INT_ST_DPHY_FATAL register (cleared on read)

41.7.4 CSI Data Identifier Register 1

Name: CSI_DATA_IDS_1
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Programs the data IDs for matching line error reporting.

Configures Data ID definition for four data types and virtual channel identifiers.

Bit	31	30	29	28	27	26	25	24
	DI3_VC[1:0]		DI3_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DI2_VC[1:0]		DI2_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DI1_VC[1:0]		DI1_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DI0_VC[1:0]		DI0_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – DI3_VC[1:0] Virtual channel for programmed data ID 3.

Bits 29:24 – DI3_DT[5:0] Data type for programmed data ID 3.

Bits 23:22 – DI2_VC[1:0] Virtual channel for programmed data ID 2.

Bits 21:16 – DI2_DT[5:0] Data type for programmed Data ID 2.

Bits 15:14 – DI1_VC[1:0] Virtual channel for programmed data ID 1.

Bits 13:8 – DI1_DT[5:0] Data type for programmed data ID 1.

Bits 7:6 – DI0_VC[1:0] Virtual channel for programmed data ID 0.

Bits 5:0 – DI0_DT[5:0] Data type for programmed data ID 0.

41.7.5 CSI Data Identifier Register 2

Name: CSI_DATA_IDS_2
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

Programs the data IDs for matching line error reporting.

Configures Data ID definition for four data types and virtual channel identifiers.

Bit	31	30	29	28	27	26	25	24
	DI7_VC[1:0]		DI7_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DI6_VC[1:0]		DI6_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DI5_VC[1:0]		DI5_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DI4_VC[1:0]		DI4_DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – DI7_VC[1:0] Virtual channel for programmed data ID 7

Bits 29:24 – DI7_DT[5:0] Data type for programmed data ID 7

Bits 23:22 – DI6_VC[1:0] Virtual channel for programmed data ID 6

Bits 21:16 – DI6_DT[5:0] Data type for programmed data ID 6

Bits 15:14 – DI5_VC[1:0] Virtual channel for programmed data ID 5

Bits 13:8 – DI5_DT[5:0] Data type for programmed data ID 5

Bits 7:6 – DI4_VC[1:0] Virtual channel for programmed data ID 4

Bits 5:0 – DI4_DT[5:0] Data type for programmed data ID 4

41.7.6 CSI D-PHY Shutdown Register

Name: CSI_DPHY_SHUTDOWNZ
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DPHY_SHUTDOWNZ
Access								R/W
Reset								0

Bit 0 – DPHY_SHUTDOWNZ Shutdown input

Value	Description
0	Puts the D-PHY in Shutdown mode. All analog blocks are in Power-down mode and digital logic is cleared.
1	Exits from Shutdown mode.

41.7.7 CSI D-PHY Reset Register

Name: CSI_DPHY_RSTZ
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DPHY_RSTZ
Access								R/W
Reset								0

Bit 0 – DPHY_RSTZ D-PHY reset control

Value	Description
0	Resets the D-PHY.
1	D-PHY exits Reset state.

41.7.8 CSI D-PHY Receive Status Register

Name: CSI_DPHY_RX
Offset: 0x48
Reset: 0x00010000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DPHY_RXCLK ACTIVEHS	DPHY_RXULP SCLKNOT
Access							R	R
Reset							0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DPHY_RXULP SESC_1	DPHY_RXULP SESC_0
Access							R	R
Reset							0	0

Bit 17 – DPHY_RXCLKACTIVEHS Active high. Indicates that D-PHY clock lane is actively receiving a DDR clock.

Bit 16 – DPHY_RXULPSCLKNOT Active low. Indicates that D-PHY Clock Lane module has entered the Ultra-Low-Power state.

Bit 1 – DPHY_RXULPSESC_1 Active high. Lane module 1 has entered Ultra-Low-Power mode

Bit 0 – DPHY_RXULPSESC_0 Active high. Lane module 0 has entered Ultra-Low-Power mode

41.7.9 CSI D-PHY Stop State Register

Name: CSI_DPHY_STOPSTATE
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								DPHY_STOPSTATECLK
Access								R
Reset								0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							DPHY_STOPSTATEDATA_1	DPHY_STOPSTATEDATA_0
Access							R	R
Reset							0	0

Bit 16 – DPHY_STOPSTATECLK Active high. D-PHY Clock lane in Stop state

Bit 1 – DPHY_STOPSTATEDATA_1 Active high. Data lane 1 in Stop state

Bit 0 – DPHY_STOPSTATEDATA_0 Active high. Data lane 0 in Stop state

41.7.10 CSI D-PHY Analog Configuration Control Register

Name: CSI_DPHY_TEST_CTRL0
Offset: 0x50
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DPHY_TESTCLK	DPHY_TESTCLR
Access							R/W	R/W
Reset							0	1

Bit 1 – DPHY_TESTCLK Analog configuration control clock

Value	Description
0	No effect.
1	Captures the DPHY_TEST_CTRL1.DPHY_TESTDIN value. **Note:** The data is loaded on one edge of DPHY_TESTCLK, thus it is mandatory to write a '0' immediately after writing a '1'. Refer to DPHY_TEST_CTRL1.DPHY_TESTDEN.

Bit 0 – DPHY_TESTCLR Analog configuration clear

Value	Description
0	No effect.
1	Resets the analog configuration. **Note:** The reset is performed on the rising edge of DPHY_TESTCLR, thus it is mandatory to write a '0' immediately after writing a '1'.

41.7.11 CSI D-PHY Analog Configuration Data Register

Name: CSI_DPHY_TEST_CTRL1
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								DPHY_TESTEN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DPHY_TESTDIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – DPHY_TESTEN Analog configuration code selection

Value	Description
0	Transmits the high-speed bit rate code on the rising edge of CSI_DPHY_TEST_CTRL0.DPHY_TESTCLK.
1	Transmits the address (0x44) of the high-speed bit rate code on the falling edge of CSI_DPHY_TEST_CTRL0.DPHY_TESTCLK.

Bits 7:0 – DPHY_TESTDIN[7:0] Data bus carrying the value selected by DPHY_TESTEN.

41.7.12 CSI D-PHY Calibration Status Register

Name: CSI_DPHY_CAL
Offset: 0xCC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								RXSKEWCALHS
Access								R
Reset								0

Bit 0 – RXSKEWCALHS A low-to-high transition on this bit means that the D-PHY has initiated the calibration. (cleared on read)

41.7.13 CSI D-PHY Fatal Error Interrupt Status Register

Name: CSI_INT_ST_DPHY_FATAL
Offset: 0xE0
Reset: 0x00000000
Property: Read-only

Interrupt sources related to loss of synchronization in the D-PHY. Packet discarded.

Reading CSI_INT_ST_DPHY_FATAL register does not clear the interrupt pin.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DPHY_ERRSOTSYNCHS_1	DPHY_ERRSOTSYNCHS_0
Access							R	R
Reset							0	0

Bit 1 – DPHY_ERRSOTSYNCHS_1 Start of transmission error on data lane 1 (no synchronization achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY_FATAL.
1	Transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY_FATAL.

Bit 0 – DPHY_ERRSOTSYNCHS_0 Start of transmission error on data lane 0 (no synchronization achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY_FATAL.
1	Transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY_FATAL.

41.7.14 CSI D-PHY Fatal Error Interrupt Mask Register

Name: CSI_INT_MSK_DPHY_FATAL
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							MASK_DPHY_ERRSOTSYNCHS_1	MASK_DPHY_ERRSOTSYNCHS_0
Access							R/W	R/W
Reset							0	0

Bit 1 – MASK_DPHY_ERRSOTSYNCHS_1 Mask for DPHY_ERRSOTSYNCHS_1

Bit 0 – MASK_DPHY_ERRSOTSYNCHS_0 Mask for DPHY_ERRSOTSYNCHS_0

41.7.15 CSI Packet Fatal Error Interrupt Status Register

Name: CSI_INT_ST_PKT_FATAL
Offset: 0xF0
Reset: 0x00000000
Property: Read-only

Notifies which interrupt bit has caused the interruption.

Reading CSI_INT_ST_PKT_FATAL does not clear the interrupt pin.

The following configuration values are valid for all listed bit names of this register:

0: No event occurred since the last read of the register.

1: An event occurred since the last read of the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ERR_ECC_DOUBLE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					VC3_ERR_CRC	VC2_ERR_CRC	VC1_ERR_CRC	VC0_ERR_CRC
Access					R	R	R	R
Reset					0	0	0	0

Bit 16 – ERR_ECC_DOUBLE D-PHY mode: Header ECC contains at least 2 errors, unrecoverable. (cleared on read)
C-PHY mode: Header CRC unrecoverable.

Bit 3 – VC3_ERR_CRC Payload Checksum error detected on virtual channel 3 (cleared on read)

Bit 2 – VC2_ERR_CRC Payload Checksum error detected on virtual channel 2 (cleared on read)

Bit 1 – VC1_ERR_CRC Payload Checksum error detected on virtual channel 1 (cleared on read)

Bit 0 – VC0_ERR_CRC Payload Checksum error detected on virtual channel 0 (cleared on read)

41.7.16 CSI Packet Fatal Error Interrupt Mask Register

Name: CSI_INT_MSK_PKT_FATAL
Offset: 0xF4
Reset: 0x00000000
Property: Read/Write

Interrupt mask for CSI_INT_ST_PKT_FATAL controls which interrupt status bits trigger the interrupt pin.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								MASK_ERR_ECC_DOUBLE
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					MASK_VC3_ERR_CRC	MASK_VC2_ERR_CRC	MASK_VC1_ERR_CRC	MASK_VC0_ERR_CRC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 16 – MASK_ERR_ECC_DOUBLE Mask for ERR_ECC_DOUBLE

Bit 3 – MASK_VC3_ERR_CRC Mask for VC3_ERR_CRC

Bit 2 – MASK_VC2_ERR_CRC Mask for VC2_ERR_CRC

Bit 1 – MASK_VC1_ERR_CRC Mask for VC1_ERR_CRC

Bit 0 – MASK_VC0_ERR_CRC Mask for VC0_ERR_CRC

41.7.17 CSI Frame Error Interrupt Status Register

Name: CSI_INT_ST_FRAME_FATAL
Offset: 0x100
Reset: 0x00000000
Property: Read-only

Interrupt sources related to Frame construction. Packet discarded.

Reading CSI_INT_ST_FRAME_FATAL does not clear the interrupt pin.

The following configuration values are valid for all listed bit names of this register:

0: No event occurred since the last read of the register.

1: An event occurred since the last read of the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					ERR_FRAME_DATA_VC3	ERR_FRAME_DATA_VC2	ERR_FRAME_DATA_VC1	ERR_FRAME_DATA_VC0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					ERR_F_SEQ_VC3	ERR_F_SEQ_VC2	ERR_F_SEQ_VC1	ERR_F_SEQ_VC0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					ERR_F_BNDRY_MATCH_VC3	ERR_F_BNDRY_MATCH_VC2	ERR_F_BNDRY_MATCH_VC1	ERR_F_BNDRY_MATCH_VC0
Access					R	R	R	R
Reset					0	0	0	0

Bit 19 – ERR_FRAME_DATA_VC3 Last received Frame in virtual channel 3, had at least one CRC error (cleared on read)

Bit 18 – ERR_FRAME_DATA_VC2 Last received Frame in virtual channel 2, had at least one CRC error (cleared on read)

Bit 17 – ERR_FRAME_DATA_VC1 Last received Frame in virtual channel 1, had at least one CRC error (cleared on read)

Bit 16 – ERR_FRAME_DATA_VC0 Last received Frame in virtual channel 0, had at least one CRC error (cleared on read)

Bit 11 – ERR_F_SEQ_VC3 Incorrect Frame sequence detected in virtual channel 3 (cleared on read)

Bit 10 – ERR_F_SEQ_VC2 Incorrect Frame sequence detected in virtual channel 2 (cleared on read)

Bit 9 – ERR_F_SEQ_VC1 Incorrect Frame sequence detected in virtual channel 1 (cleared on read)

Bit 8 – ERR_F_SEQ_VC0 Incorrect Frame sequence detected in virtual channel 0 (cleared on read)

Bit 3 – ERR_F_BNDRY_MATCH_VC3 Error matching Frame Start with Frame End for virtual channel 3 (cleared on read)

Bit 2 – ERR_F_BNDRY_MATCH_VC2 Error matching Frame Start with Frame End for virtual channel 2 (cleared on read)

Bit 1 – ERR_F_BNDRY_MATCH_VC1 Error matching Frame Start with Frame End for virtual channel 1 (cleared on read)

Bit 0 – ERR_F_BNDRY_MATCH_VC0 Error matching Frame Start with Frame End for virtual channel 0 (cleared on read)

41.7.18 CSI Frame Error Interrupt Mask Register

Name: CSI_INT_MSK_FRAME_FATAL
Offset: 0x104
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MASK_ERR_FRAME_DATA_VC3	MASK_ERR_FRAME_DATA_VC2	MASK_ERR_FRAME_DATA_VC1	MASK_ERR_FRAME_DATA_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					MASK_ERR_F_SEQ_VC3	MASK_ERR_F_SEQ_VC2	MASK_ERR_F_SEQ_VC1	MASK_ERR_F_SEQ_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MASK_ERR_F_BNDRY_MATCH_VC3	MASK_ERR_F_BNDRY_MATCH_VC2	MASK_ERR_F_BNDRY_MATCH_VC1	MASK_ERR_F_BNDRY_MATCH_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – MASK_ERR_FRAME_DATA_VC3 Mask for ERR_FRAME_DATA_VC3

Bit 18 – MASK_ERR_FRAME_DATA_VC2 Mask for ERR_FRAME_DATA_VC2

Bit 17 – MASK_ERR_FRAME_DATA_VC1 Mask for ERR_FRAME_DATA_VC1

Bit 16 – MASK_ERR_FRAME_DATA_VC0 Mask for ERR_FRAME_DATA_VC0

Bit 11 – MASK_ERR_F_SEQ_VC3 Mask for ERR_F_SEQ_VC3

Bit 10 – MASK_ERR_F_SEQ_VC2 Mask for ERR_F_SEQ_VC2

Bit 9 – MASK_ERR_F_SEQ_VC1 Mask for ERR_F_SEQ_VC1

Bit 8 – MASK_ERR_F_SEQ_VC0 Mask for ERR_F_SEQ_VC0

Bit 3 – MASK_ERR_F_BNDRY_MATCH_VC3 Mask for ERR_F_BNDRY_MATCH_VC3

Bit 2 – MASK_ERR_F_BNDRY_MATCH_VC2 Mask for ERR_F_BNDRY_MATCH_VC2

Bit 1 – MASK_ERR_F_BNDRY_MATCH_VC1 Mask for ERR_F_BNDRY_MATCH_VC1

Bit 0 – MASK_ERR_F_BNDRY_MATCH_VC0 Mask for ERR_F_BNDRY_MATCH_VC0

41.7.19 CSI D-PHY Interrupt Status Register

Name: CSI_INT_ST_DPHY
Offset: 0x110
Reset: 0x00000000
Property: Read-only

Interrupt caused by the D-PHY.

Reading CSI_INT_ST_DPHY does not clear the interrupt pin.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DPHY_ERRESC_1	DPHY_ERRESC_0
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DPHY_ERRSOTHS_1	DPHY_ERRSOTHS_0
Access							R	R
Reset							0	0

Bit 17 – DPHY_ERRESC_1 Start of transmission error on data lane 1 (no synchronization achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY.
1	Transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY.

Bit 16 – DPHY_ERRESC_0 Start of transmission error on data lane 0 (no synchronization achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY.
1	Transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY.

Bit 1 – DPHY_ERRSOTHS_1 Start of transmission error on data lane 1 (synchronization can still be achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY.
1	Transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_DPHY.

Bit 0 – DPHY_ERRSOTHS_0 Start of transmission error on data lane 0 (synchronization can still be achieved) (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY.
1	Transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_DPHY.

41.7.20 CSI D-PHY Interrupt Mask Register

Name: CSI_INT_MSK_DPHY
Offset: 0x114
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							MASK_DPHY_ERRESC_1	MASK_DPHY_ERRESC_0
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							MASK_DPHY_ERRSOTHS_1	MASK_DPHY_ERRSOTHS_0
Access							R/W	R/W
Reset							0	0

Bit 17 – MASK_DPHY_ERRESC_1 Mask for DPHY_ERRESC_1

Bit 16 – MASK_DPHY_ERRESC_0 Mask for DPHY_ERRESC_0

Bit 1 – MASK_DPHY_ERRSOTHS_1 Mask for DPHY_ERRSOTHS_1

Bit 0 – MASK_DPHY_ERRSOTHS_0 Mask for DPHY_ERRSOTHS_0

41.7.21 CSI Virtual Channel Interrupt Status Register

Name: CSI_INT_ST_PKT
Offset: 0x120
Reset: 0x00000000
Property: Read-only

Interrupt related to Packet construction. Packet discarded.

Reading CSI_INT_ST_PKT does not clear the interrupt pin.

The following configuration values are valid for all listed bit names of this register:

0: No event occurred since the last read of the register.

1: An event occurred since the last read of the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					VC3_ERR_ECC_CORRECTED	VC2_ERR_ECC_CORRECTED	VC1_ERR_ECC_CORRECTED	VC0_ERR_ECC_CORRECTED
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ERR_ID_VC3	ERR_ID_VC2	ERR_ID_VC1	ERR_ID_VC0
Access					R	R	R	R
Reset					0	0	0	0

Bit 19 – VC3_ERR_ECC_CORRECTED D-PHY mode: Header error detected and corrected on virtual channel 3 (cleared on read)
C-PHY mode: Header CRC recoverable on virtual channel 3

Bit 18 – VC2_ERR_ECC_CORRECTED D-PHY mode: Header error detected and corrected on virtual channel 2 (cleared on read)
C-PHY mode: Header CRC recoverable on virtual channel 2

Bit 17 – VC1_ERR_ECC_CORRECTED D-PHY mode: Header error detected and corrected on virtual channel 1 (cleared on read)
C-PHY mode: Header CRC recoverable on virtual channel 1

Bit 16 – VC0_ERR_ECC_CORRECTED D-PHY mode: Header error detected and corrected on virtual channel 0 (cleared on read)
C-PHY mode: Header CRC recoverable on virtual channel 0

Bit 3 – ERR_ID_VC3 Unrecognized or unimplemented data type detected in virtual channel 3 (cleared on read)

Bit 2 – ERR_ID_VC2 Unrecognized or unimplemented data type detected in virtual channel 2 (cleared on read)

Bit 1 – ERR_ID_VC1 Unrecognized or unimplemented data type detected in virtual channel 1 (cleared on read)

Bit 0 – ERR_ID_VC0 Unrecognized or unimplemented data type detected in virtual channel 0 (cleared on read)

41.7.22 CSI Virtual Channel Interrupt Mask Register

Name: CSI_INT_MSK_PKT
Offset: 0x124
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MASK_VC3_ERR_ECC_CORRECTED	MASK_VC2_ERR_ECC_CORRECTED	MASK_VC1_ERR_ECC_CORRECTED	MASK_VC0_ERR_ECC_CORRECTED
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					MASK_ERR_ID_VC3	MASK_ERR_ID_VC2	MASK_ERR_ID_VC1	MASK_ERR_ID_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – MASK_VC3_ERR_ECC_CORRECTED Mask for CSI_INT_ST_PKT.VC3_ERR_ECC_CORRECTED interrupt source

Bit 18 – MASK_VC2_ERR_ECC_CORRECTED Mask for CSI_INT_ST_PKT.VC2_ERR_ECC_CORRECTED interrupt source

Bit 17 – MASK_VC1_ERR_ECC_CORRECTED Mask for CSI_INT_ST_PKT.VC1_ERR_ECC_CORRECTED interrupt source

Bit 16 – MASK_VC0_ERR_ECC_CORRECTED Mask for CSI_INT_ST_PKT.VC0_ERR_ECC_CORRECTED interrupt source

Bit 3 – MASK_ERR_ID_VC3 Mask for CSI_INT_ST_PKT.ERR_ID_VC3 interrupt source

Bit 2 – MASK_ERR_ID_VC2 Mask for CSI_INT_ST_PKT.ERR_ID_VC2 interrupt source

Bit 1 – MASK_ERR_ID_VC1 Mask for CSI_INT_ST_PKT.ERR_ID_VC1 interrupt source

Bit 0 – MASK_ERR_ID_VC0 Mask for CSI_INT_ST_PKT.ERR_ID_VC0 interrupt source

41.7.23 CSI Line Error Interrupt Status Register

Name: CSI_INT_ST_LINE
Offset: 0x130
Reset: 0x00000000
Property: Read-only

Interrupt related to Line construction.

Reading CSI_INT_ST_LINE register does not clear the interrupt pin.

The following configuration values are valid for all listed bit names of this register:

0: No event occurred since the last read of the register.

1: An event occurred since the last read of the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ERR_L_SEQ_DI7	ERR_L_SEQ_DI6	ERR_L_SEQ_DI5	ERR_L_SEQ_DI4	ERR_L_SEQ_DI3	ERR_L_SEQ_DI2	ERR_L_SEQ_DI1	ERR_L_SEQ_DI0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERR_L_BNDRY_MATCH_DI7	ERR_L_BNDRY_MATCH_DI6	ERR_L_BNDRY_MATCH_DI5	ERR_L_BNDRY_MATCH_DI4	ERR_L_BNDRY_MATCH_DI3	ERR_L_BNDRY_MATCH_DI2	ERR_L_BNDRY_MATCH_DI1	ERR_L_BNDRY_MATCH_DI0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 23 – ERR_L_SEQ_DI7 Error in the sequence of lines for vc7 and dt7 (cleared on read)

Bit 22 – ERR_L_SEQ_DI6 Error in the sequence of lines for vc6 and dt6 (cleared on read)

Bit 21 – ERR_L_SEQ_DI5 Error in the sequence of lines for vc5 and dt5 (cleared on read)

Bit 20 – ERR_L_SEQ_DI4 Error in the sequence of lines for vc4 and dt4 (cleared on read)

Bit 19 – ERR_L_SEQ_DI3 Error in the sequence of lines for vc3 and dt3 (cleared on read)

Bit 18 – ERR_L_SEQ_DI2 Error in the sequence of lines for vc2 and dt2 (cleared on read)

Bit 17 – ERR_L_SEQ_DI1 Error in the sequence of lines for vc1 and dt1 (cleared on read)

Bit 16 – ERR_L_SEQ_DI0 Error in the sequence of lines for vc0 and dt0 (cleared on read)

Bit 7 – ERR_L_BNDRY_MATCH_DI7 Error matching line start with line end for vc7 and dt7 (cleared on read)

Bit 6 – ERR_L_BNDRY_MATCH_DI6 Error matching line start with line end for vc6 and dt6 (cleared on read)

Bit 5 – ERR_L_BNDRY_MATCH_DI5 Error matching line start with line end for vc5 and dt5 (cleared on read)

Bit 4 – ERR_L_BNDRY_MATCH_DI4 Error matching line start with line end for vc4 and dt4 (cleared on read)

Bit 3 – ERR_L_BNDRY_MATCH_DI3 Error matching line start with line end for vc3 and dt3 (cleared on read)

Bit 2 – ERR_L_BNDRY_MATCH_DI2 Error matching line start with line end for vc2 and dt2 (cleared on read)

Bit 1 – ERR_L_BNDRY_MATCH_DI1 Error matching line start with line end for vc1 and dt1 (cleared on read)

Bit 0 – ERR_L_BNDRY_MATCH_DI0 Error matching line start with line end for vc0 and dt0 (cleared on read)

41.7.24 CSI Line Interrupt Mask Register

Name: CSI_INT_MSK_LINE
Offset: 0x134
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MASK_ERR_L_SEQ_DI7	MASK_ERR_L_SEQ_DI6	MASK_ERR_L_SEQ_DI5	MASK_ERR_L_SEQ_DI4	MASK_ERR_L_SEQ_DI3	MASK_ERR_L_SEQ_DI2	MASK_ERR_L_SEQ_DI1	MASK_ERR_L_SEQ_DI0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	MASK_ERR_L_BNDRY_MATCH_DI7	MASK_ERR_L_BNDRY_MATCH_DI6	MASK_ERR_L_BNDRY_MATCH_DI5	MASK_ERR_L_BNDRY_MATCH_DI4	MASK_ERR_L_BNDRY_MATCH_DI3	MASK_ERR_L_BNDRY_MATCH_DI2	MASK_ERR_L_BNDRY_MATCH_DI1	MASK_ERR_L_BNDRY_MATCH_DI0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 23 – MASK_ERR_L_SEQ_DI7 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI7 interrupt source

Bit 22 – MASK_ERR_L_SEQ_DI6 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI6 interrupt source

Bit 21 – MASK_ERR_L_SEQ_DI5 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI5 interrupt source

Bit 20 – MASK_ERR_L_SEQ_DI4 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI4 interrupt source

Bit 19 – MASK_ERR_L_SEQ_DI3 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI3 interrupt source

Bit 18 – MASK_ERR_L_SEQ_DI2 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI2 interrupt source

Bit 17 – MASK_ERR_L_SEQ_DI1 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI1 interrupt source

Bit 16 – MASK_ERR_L_SEQ_DI0 Mask for CSI_INT_ST_LINE.ERR_L_SEQ_DI0 interrupt source

Bit 7 – MASK_ERR_L_BNDRY_MATCH_DI7 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI7 interrupt source

Bit 6 – MASK_ERR_L_BNDRY_MATCH_DI6 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI6 interrupt source

Bit 5 – MASK_ERR_L_BNDRY_MATCH_DI5 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI5 interrupt source

Bit 4 – MASK_ERR_L_BNDRY_MATCH_DI4 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI4 interrupt source

Bit 3 – MASK_ERR_L_BNDRY_MATCH_DI3 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI3 interrupt source

Bit 2 – MASK_ERR_L_BNDRY_MATCH_DI2 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI2 interrupt source

Bit 1 – MASK_ERR_L_BNDRY_MATCH_DI1 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI1 interrupt source

Bit 0 – MASK_ERR_L_BNDRY_MATCH_DI0 Mask for CSI_INT_ST_LINE.ERR_L_BNDRY_MATCH_DI0 interrupt source

41.7.25 CSI Descrambler Configuration Register

Name: CSI_SCRAMBLING
Offset: 0x300
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								SCRAMBLE_ENABLE
Access								R/W
Reset								0

Bit 0 – SCRAMBLE_ENABLE Active high. Enables data descrambling in the CSI

41.7.26 CSI Lane 0 Scrambling Seed Register

Name: CSI_SCRAMBLING_SEED0
Offset: 0x304
Reset: 0x00001008
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SCRAMBLE_SEED_LANE0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SCRAMBLE_SEED_LANE0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bits 15:0 – SCRAMBLE_SEED_LANE0[15:0] Seed used by the descrambler block for data lane 0

41.7.27 CSI Lane 1 Scrambling Seed Register

Name: CSI_SCRAMBLING_SEED1
Offset: 0x308
Reset: 0x00001188
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SCRAMBLE_SEED_LANE1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	SCRAMBLE_SEED_LANE1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	0	0	0

Bits 15:0 – SCRAMBLE_SEED_LANE1[15:0] Seed used by the descrambler block for data lane 1

42. CSI-2 Demultiplexer Controller (CSI2DC)

42.1 Description

The CSI-2 Demultiplexer Controller (CSI2DC) receives incoming data from a CSI-2 physical interface and filters packets based on their data type and virtual channel identifier.

The CSI2DC is fed with four clock domains — the CSI-2 data clock domain, the ISC clock domain, the configuration clock domain and the system bus clock domain. The CSI2DC performs clock domain crossing of incoming packets.

The CSI2DC integrates one video pipeline and one data pipeline. The video pipeline converts the byte stream to a pixel stream with an optional RAW decompression algorithm. This pipeline is connected to the Image Sensor Controller (ISC). The data pipeline propagates the data packets to a system bus client interface. The data buffers are then processed by the processor or by the centralized Direct Memory Access (DMA) controller using DMA requests.

The CSI2DC also includes a snoop controller that captures image data and packet attributes helping system bring up and debug.

The table below summarizes acronyms used in the following sections.

Table 42-1. Acronyms Used

CDC	Clock Domain Crossing
CSI	Camera Serial Interface
D-PHY	D Physical Layer
DMA	Direct Memory Access
DT	Data Type
FE	Frame End
FIFO	First In First Out
FS	Frame Start
GS	Generic Short
HS	High Speed; identifier for operation mode
ISC	Image Sensor Controller
ISP	Image Signal Processor
LE	Line End
LP	Low Power; identifier for operation mode
LS	Line Start
PFE	Parallel Front End
PHY	Physical Layer
RC	Row Count
RGB	Color representation (Red, Green, Blue)
SSP	Synchronization Short Packet
SW	Software
VC	Virtual Channel
WC	Word Count

YUV	Color representation (Y for luminance, U & V for chrominance)

42.2 Embedded Characteristics

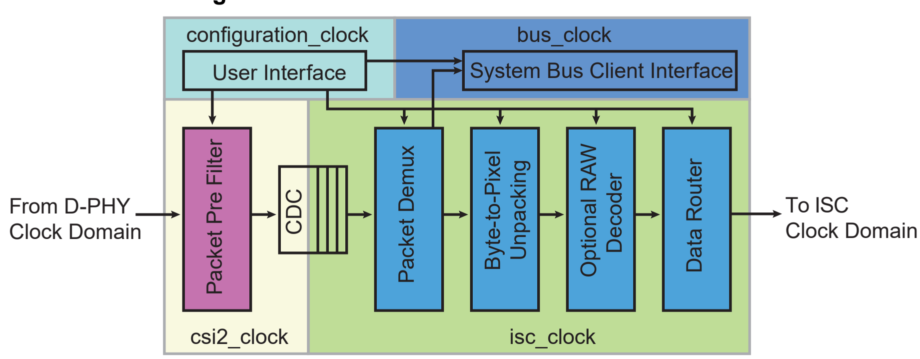
- CSI-2 Version 1.3 Specification-compliant Demultiplexer
- Four Virtual Channels
- Primary and Secondary Image Format
- RGB, YUV, RAW, Compressed RAW, User-defined Byte-based Packets
- Recommended Memory Storage Format or Single Pixel Per Clock Cycle
- Non-Image Data Packet and Generic Short Packet Support
- Image Data Snoop Controller (four entries)
- Programmable Video Pipeline Filter
- Programmable Data Pipeline Filter
- Progressive and Interlaced Content
- Programmable DMA Client Interface

42.3 Block Diagram

The CSI2DC includes four clocks:

- CSI-2 clock domain (CSI output): to sample the incoming byte stream along with protocol signals; up to 4 bytes per cycle are sampled.
- ISC clock domain (ISC MCLK output): to extract pixels from the data packet; one pixel per clock cycle is retrieved in RGB or RAW mode.
- Configuration clock domain: for configuration and status
- System bus clock domain: to read data from the RAM

Figure 42-1. CSI2DC Block Diagram



42.4 I/O Lines Description

Table 42-2. I/O Lines Description

Signal Name	Description	Type
ISC_PCK	Image Sensor Pixel clock	Input
ISC_D[11:0]	Image Sensor Data	Input
ISC_VSYNC	Image Sensor Vertical Synchro	Input
ISC_HSYNC	Image Sensor Horizontal Synchro	Input
ISC_FIELD	Field Identification Signal	Input

...........continued

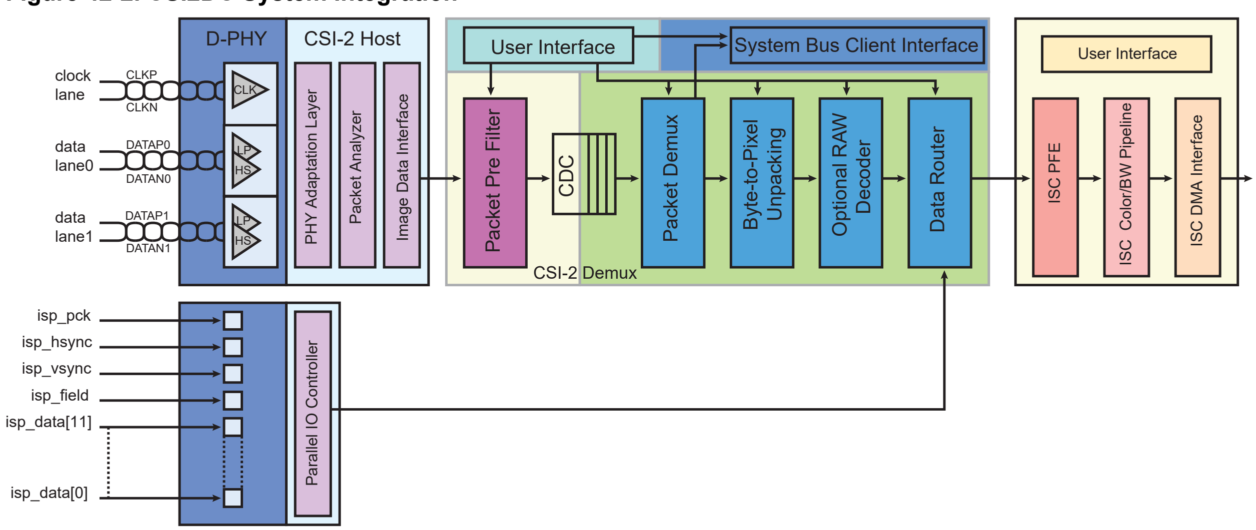
Signal Name	Description	Type
ISC_MCK	Image Sensor Main clock	Output
MIPI_CLKP	MIPI DPHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI DPHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI DPHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_DP2	MIPI DPHY differential output data lane 2	Input/Output
MIPI_DN2		
MIPI_DP3	MIPI DPHY differential output data lane 3	Input/Output
MIPI_DN3		
MIPI_REXT	Calibration reference resistor	Input/Output

42.5 Functional Description

As shown in the following figure, the CSI2DC is connected to a CSI-2 protocol host side. The host side receives data from a CSI-2-compliant camera. The supported host protocol, Image Data Interface (IDI), uses a 32-bit data bus, vertical and horizontal timing accurate video synchronization signals, data type and virtual channel.

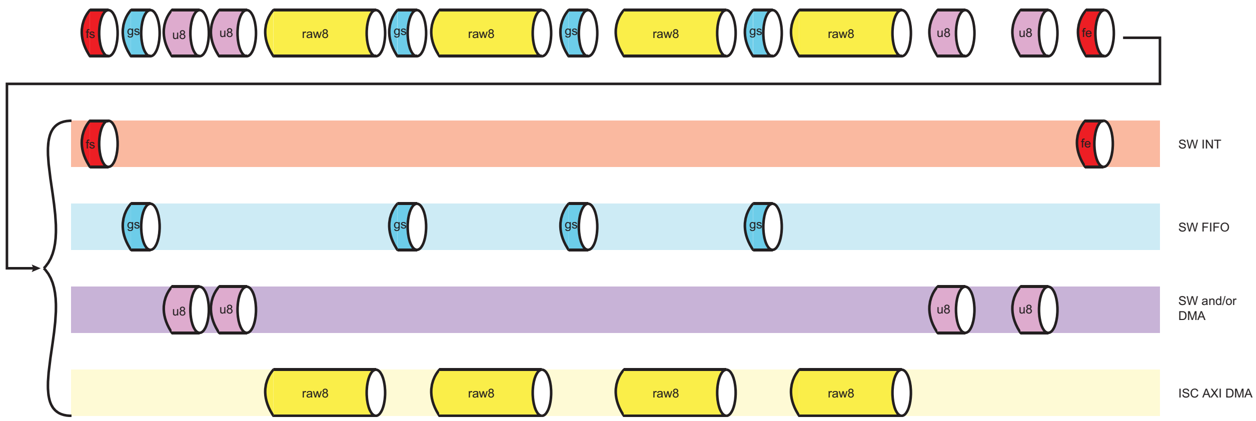
The CSI2DC receives a packet-based data stream, and outputs a pixel stream through its video pipeline connected to the Image Sensor Controller.

Figure 42-2. CSI2DC System Integration



A system bus client port is also available to retrieve non-image data that are locally saved into buffers. The dual buffer operation can be used with centralized DMA assistance or by using CPU accesses. When a packet is received, a snoop controller can be used to monitor CSI-2 traffic and identify that the desired information is successfully transmitted and received in the CSI2DC. Once configured, the video pipe filters packets and transmits a predefined image data type. Only relevant packets are forwarded to the ISC clock domain. A buffer is used to adjust clock domain crossing between the CSI-2 host domain and the ISC clock domain. A simple queue is used to pass data packets from the CSI-2 host clock domain to the system bus clock domain. The packet is routed based on its data type. The video pipe must be enabled to route data from the camera to the ISC. The data pipe must be enabled to route data from the camera to the system bus client interface.

Figure 42-3. Packet Demultiplexing Example



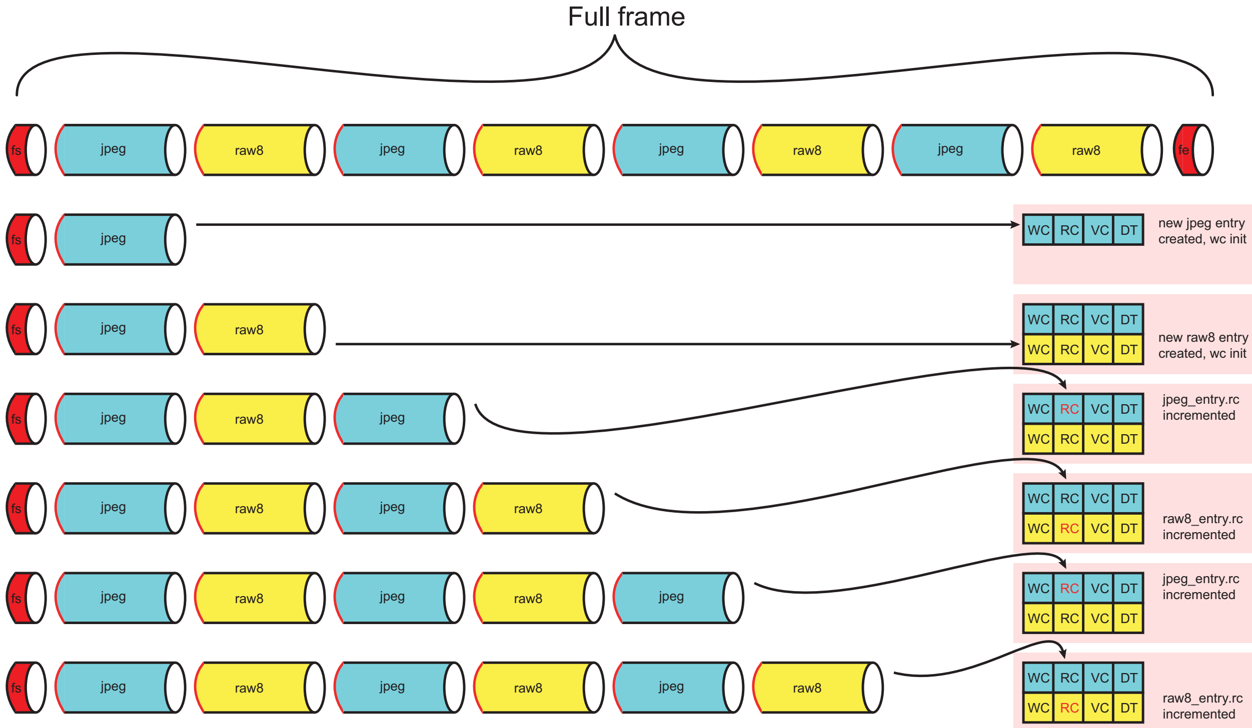
42.5.1 Image Data Snoop (IDS) Controller

The Image Data Snoop Controller allows packet monitoring for system-level bring up and debug. When a new image data packet is received, a new entry is created. An entry is composed of the data type, the virtual channel identifier, the word count and the row count. When a new packet with the same virtual channel identifier and data type is received, the row count field is incremented. The table of entry is transferred at the end to the user interface for processor analysis.

Table 42-3. CSI2DC Image Data Snoop Controller

Packet	Data Type	Image Data Snoop Support
Generic Short Packet (GSP)	GSP code 1 to 8	No
Generic Long Packet (GLP)	Null or blanking data	Interrupt
	Embedded 8-bit non-image data	Yes
	Reserved	GLP error interrupt reserved
YUV Image Data	YUV	Yes
	Reserved	YUV error interrupt
RGB Image Data	RGB	Yes
	Reserved	RGB error interrupt
RAW Image Data	RAW 6 to 14	Yes
	Reserved	RAW error interrupt
User-defined 8-bit Data	User-defined 8-bit data types 1 to 8	Yes

Figure 42-4. Image Data Snoop Controller



42.5.2 Synchronization Short Packet Demux

Short packets include Frame Synchronization packets and Line Synchronization packets. Each image must begin with a Frame Start (FS) Packet containing the Frame Start Code, and must end with a Frame End (FE) packet containing the Frame End code.

Table 42-4. Synchronization Short Packet Demultiplexing

Packet	Data Type	Handler
Synchronization Short Packet (SSP)	Frame Start Code	Per Virtual Channel Interrupt
	Frame End Code	Per Virtual Channel Interrupt
	Line Start Code (optional)	Per Virtual Channel Interrupt
	Line End Code (optional)	Per Virtual Channel Interrupt
	Reserved	SSP Error Interrupt reserved

The 16-bit frame number can be retrieved in the CSI2DC user interface. The Short Packet Data field indicates the frame number or the line number depending on the configuration. The behavior of the 16-bit frame number must be one of the following:

- Frame number is always 0, meaning that the frame number is inoperative.
- Frame number increments by 1 for every FS packet received with the same virtual channel identifier and is periodically reset to one, e.g. 1, 2, 1, 2, 1, 2 or 1, 2, 3, 4, 1, 2, 3, 4.
- The Frame number must be a non-value.

The behavior of the 16-bit line number must be one of the following:

- Line number is always zero — line number is inoperative.
- Line number increments by one for every Line Start (LS) packet received within the same virtual channel and the same data type. The line number is periodically reset to one for the first LS packet after a FS packet. The intended usage is for progressive scan.

- Line number increments by same arbitrary step value greater than one for every LS packet with the same virtual channel and the same data type. The line number is periodically reset to a non-zero arbitrary start value for the first LS after an FS packet. The arbitrary start value may be different between successive frames. The intended usage is for interlaced video data.

42.5.3 Generic Short Packet Demux

Generic Short Packets use the data type range [0x08, 0x0F]. These packets are retrieved through the CSI2DC user interface queue. An interrupt is raised as soon as the packet is received. The queue depth is set to 4 words. The data field retrieved is 22 bits wide and contains the 6-bit data type and a 16-bit data field. These values are passed to the application layer.

Table 42-5. Generic Short Packet Demultiplexing

Packet	Data Type	Handler
Generic Short Packet (GSP)	Code 1	Per Virtual Channel GSP queue
	Code 2	Per Virtual Channel GSP queue
	Code 3	Per Virtual Channel GSP queue
	Code 4	Per Virtual Channel GSP queue
	Code 5	Per Virtual Channel GSP queue
	Code 6	Per Virtual Channel GSP queue
	Code 7	Per Virtual Channel GSP queue
	Code 8	Per Virtual Channel GSP queue

42.5.4 Generic Long Packet Demux

The generic long packet contains three packet subclasses:

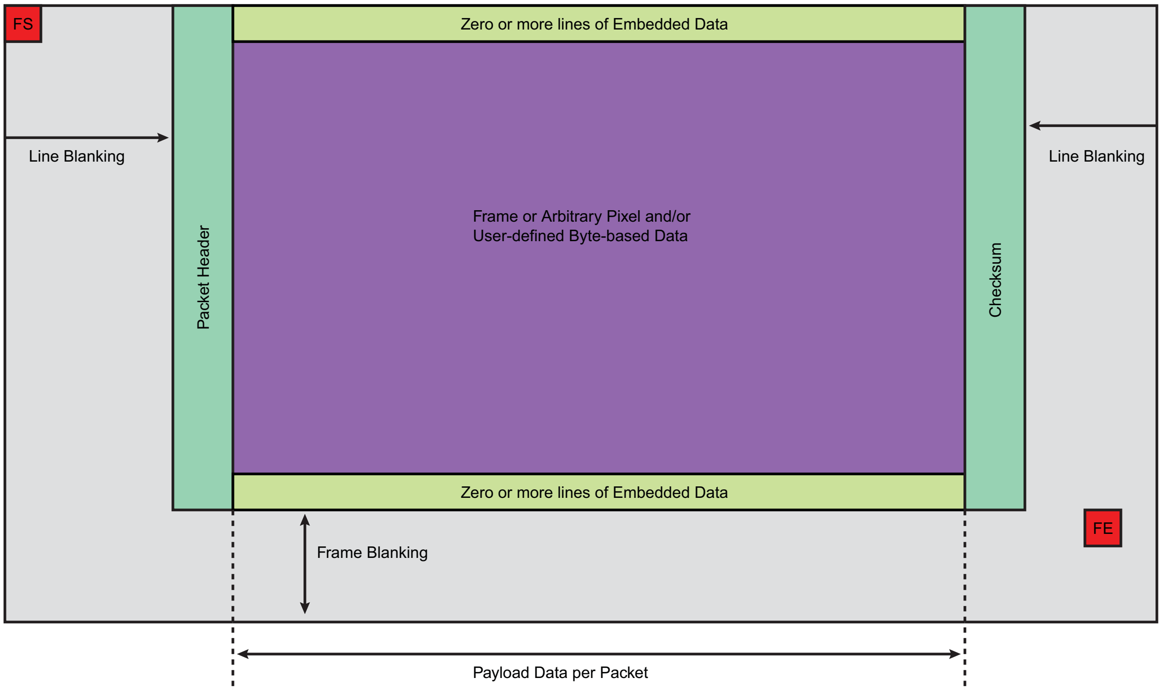
- NULL packet, data type set to 0x10
- Blanking data, data type set to 0x11
- Embedded 8-bit non-image data, data type set to 0x12

Null and blanking data packet content must be ignored by the application layer. Embedded 8-bit non-image data packets can be used to forward additional information from the camera to the host processor. These packets can be routed to the data pipe interface and read by the DMA interface or processor. The data payload must be a multiple of 8 bits.

Table 42-6. Generic Long Packet Demultiplexing

Packet	Data Type	Handler
Generic Long Packet (GLP)	Null	Null interrupt
	Blanking data	Data pipe
	Embedded 8-bit non-image data	Data pipe
	Reserved	GLP Error interrupt reserved

Figure 42-5. Generic Long Packet Embedded Information Packet



42.5.5 YUV Packet Demux

YUV data packets are always retrieved using the video pipe. CSI2DC reformats the stream to be compliant with the ISC video pipeline.

Table 42-7. YUV Image Data Demultiplexing

Packet	Data Type	Handler
YUV Image Data	YUV420 8-bit	Video pipe
	YUV420 10-bit	Video pipe
	Legacy YUV420 8-bit	Video pipe
	Reserved	YUV error interrupt
	YUV420 8-bit Chroma Shifted	Video pipe
	YUV420 10-bit Chroma Shifted	Video pipe
	YUV422 8-bit	Video pipe
	YUV422 10-bit	Video pipe

42.5.5.1 YUV 420 8-bit Legacy Mode

YUV 420 legacy data type is set to 0x1A.

Figure 42-6. YUV 420 8-bit Legacy Mapping

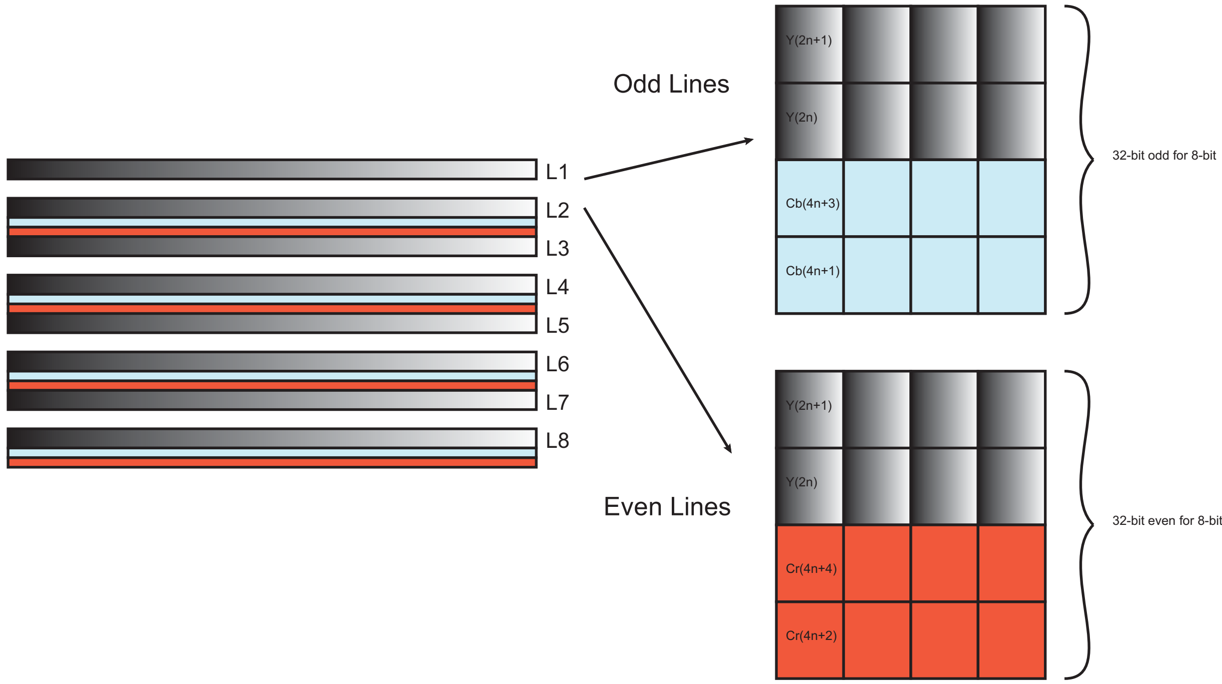


Table 42-8. YUV420 8-bit (Legacy) Odd Line Format (1, 3, 5, etc.) Mapping

DEMUX_DATA Slice	YUV 420 8-Bit (Legacy) Data Mapping
demux_data[39:30]	{Y2[7:0], 2'b00}
demux_data[29:20]	{Y1[7:0], 2'b00}
demux_data[19:10]	{U1[7:0], 2'b00} (Cb component)
demux_data[9:0]	{U3[7:0], 2'b00} (Cb component)

Table 42-9. YUV420 8-bit (Legacy) Even Line Format (2, 4, 6, etc.) Mapping

DEMUX_DATA Slice	YUV 420 8-Bit (Legacy) Data Mapping
demux_data[39:30]	{Y2[7:0], 2'b00}
demux_data[29:20]	{Y1[7:0], 2'b00}
demux_data[19:10]	{V1[7:0], 2'b00} (Cr component)
demux_data[9:0]	{V3[7:0], 2'b00} (Cr component)

Table 42-10. YUV 8-bit Legacy Recommended Memory Storage (Odd Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	U1								Y1								Y2								U3							
0x00	a7	a6	a5	a4	a3	a2	a1	a0	b7	b6	b5	b4	b3	b2	b1	b0	c7	c6	c5	c4	c3	c3	c1	c0	d7	d6	d5	d4	d3	d2	d1	d0
Pixel ID	Y3								Y4								U5								Y5							
0x04	e7	e6	e5	e4	e3	e2	e1	e0	f7	f6	f5	f4	f3	f2	f1	f0	g7	g6	g5	g4	g3	g2	g1	g0	h7	h6	h5	h4	h3	h2	h1	h0

Table 42-11. YUV 8-bit Legacy Recommended Memory Storage (Even Line)

Addr	receive_buffer[31:24]	receive_buffer[23:16]	receive_buffer[15:8]	receive_buffer[7:0]
Pixel ID	V1	Y1	Y2	V3

...........continued

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
0x00	a7	a6	a5	a4	a3	a2	a1	a0	b7	b6	b5	b4	b3	b2	b1	b0	c7	c6	c5	c4	c3	c3	c1	c0	d7	d6	d5	d4	d3	d2	d1	d0
Pixel ID	Y3								Y4								V5								Y5							
0x04	e7	e6	e5	e4	e3	e2	e1	e0	f7	f6	f5	f4	f3	f2	f1	f0	g7	g6	g5	g4	g3	g2	g1	g0	h7	h6	h5	h4	h3	h2	h1	h0

42.5.5.2 YUV 420 8-bit Mode

Data type is set to 0x18 or 0x1C.

Figure 42-7. YUV 420 8-bit Mode

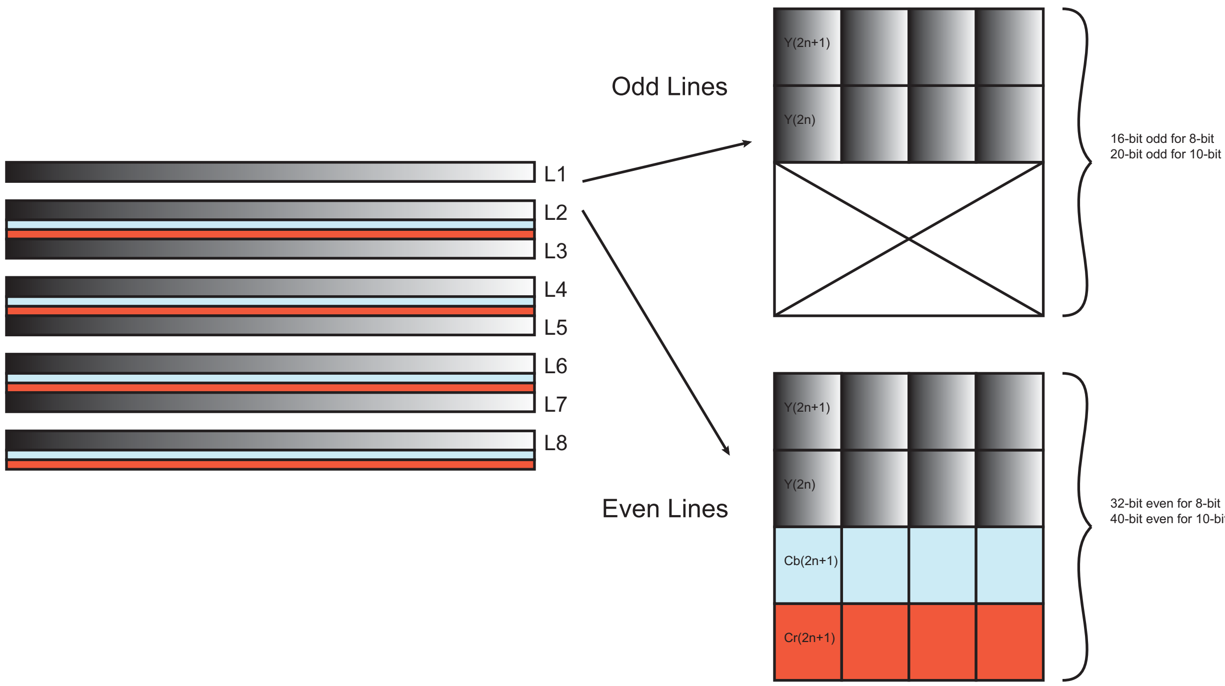


Table 42-12. YUV420 8-bit Odd Line Format Mapping

DEMUX_DATA Slice	YUV 420 8-Bit Data Mapping
demux_data[39:30]	{Y2[7:0], 2'b00}
demux_data[29:20]	{Y1[7:0], 2'b00}
demux_data[19:10]	Invalid data
demux_data[9:0]	Invalid data

Table 42-13. YUV 420 8-bit Even Line Format Mapping

DEMUX_DATA Slice	YUV 420 8-Bit Data Mapping
demux_data[39:30]	{Y2[7:0], 2'b00}
demux_data[29:20]	{Y1[7:0], 2'b00}
demux_data[19:10]	{U1[7:0], 2'b00} (Cb component)
demux_data[9:0]	{V1[7:0], 2'b00} (Cr component)

Table 42-14. YUV420 8-bit Recommended Memory Storage (Odd Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	Y4								Y3								Y2								Y1							
0x00	d7	d6	d5	d4	d3	d2	d1	d0	c7	c6	c5	c4	c3	c2	c1	c0	b7	b6	b5	b4	b3	b2	b1	b0	a7	a6	a5	a4	a3	a2	a1	a0
Pixel ID	Y8								Y7								Y6								Y5							
0x04	h7	h6	h5	h4	h3	h2	h1	h0	g7	g6	g5	g4	g3	g2	g1	g0	f7	f6	f5	f4	f3	f2	f1	f0	e7	e6	e5	e4	e3	e2	e1	e0

Table 42-15. YUV420 8-bit Recommended Memory Storage (Even Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	Y2								V1								Y1								U1							
0x00	d7	d6	d5	d4	d3	d2	d1	d0	c7	c6	c5	c4	c3	c2	c1	c0	b7	b6	b5	b4	b3	b2	b1	b0	a7	a6	a5	a4	a3	a2	a1	a0
Pixel ID	Y4								V3								Y3								U3							
0x04	h7	h6	h5	h4	h3	h2	h1	h0	g7	g6	g5	g4	g3	g2	g1	g0	f7	f6	f5	f4	f3	f2	f1	f0	e7	e6	e5	e4	e3	e2	e1	e0

42.5.5.3 YUV 420 10-bit Mode

Data type is set to 0x19 or 0x1D.

Table 42-16. YUV420 10-bit Odd Line Format Mapping

DEMUX_DATA Slice	YUV 420 10-Bit Data Mapping
demux_data[39:30]	Y2[9:0]
demux_data[29:20]	Y1[9:0]
demux_data[19:10]	Invalid data
demux_data[9:0]	Invalid data

Table 42-17. YUV 420 10-bit Even Line Format Mapping

DEMUX_DATA Slice	YUV 420 10-Bit Data mapping
demux_data[39:30]	Y2[9:0]
demux_data[29:20]	Y1[9:0]
demux_data[19:10]	U1[9:0] (Cb component)
demux_data[9:0]	V1[9:0](Cr component)

Table 42-18. YUV420 10-bit Recommended Memory Storage (Odd Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	Y4[9:2]								Y3[9:2]								Y2[9:2]								Y1[9:2]							
0x00	d9	d8	d7	d6	d5	d4	d3	d2	c9	c8	c7	c6	c5	c4	c3	c2	b9	b8	b7	b6	b5	b4	b3	b2	a9	a8	a7	a6	a5	a4	a3	a2
Pixel ID	Y7[9:2]								Y6[9:2]								Y5[9:2]								Y4[1:0]		Y3[1:0]		Y2[1:0]		Y1[1:0]	
0x04	g9	g8	g7	g6	g5	g4	g3	g2	f9	f8	f7	f6	f5	f4	f3	d2	e9	e8	e7	e6	e5	e4	e3	e2	d1	d0	c1	c0	b1	b0	a1	a0
Pixel ID	Y10[9:2]								Y9[9:2]								Y8[1:0]		Y7[1:0]		Y6[1:0]		Y5[1:0]		Y8[9:2]							
0x08	j9	j8	j7	j6	j5	j4	j3	j2	i9	i8	i7	i6	i5	i4	i3	i2	h1	h0	g1	g0	f1	f0	e1	e0	h9	h8	h7	h6	h5	h4	h3	h2

Table 42-19. YUV420 10-bit Recommended Memory Storage (Even Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	Y2[9:2]								V1[9:2]								Y1[9:2]								U1[9:2]							
0x00	d9	d8	d7	d6	d5	d4	d3	d2	c9	c8	c7	c6	c5	c4	c3	c2	b9	b8	b7	b6	b5	b4	b3	b2	a9	a8	a7	a6	a5	a4	a3	a2
Pixel ID	V3[9:2]								Y3[9:2]								U3[9:2]								Y2[1:0]		V1[1:0]		Y1[1:0]		U1[1:0]	

...........continued

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
0x04	g9	g8	g7	g6	g5	g4	g3	g2	f9	f8	f7	f6	f5	f4	f3	d2	e9	e8	e7	e6	e5	e4	e3	e2	d1	d0	c1	c0	b1	b0	a1	a0
Pixel ID	Y5[9:2]								U5[9:2]								Y4[1:0]		V3[1:0]		Y3[1:0]		U3[1:0]		Y4[9:2]							
0x08	j9	j8	j7	j6	j5	j4	j3	j2	i9	i8	i7	i6	i5	i4	i3	i2	h1	h0	g1	g0	f1	f0	e1	e0	h9	h8	h7	h6	h5	h4	h3	h2

42.5.5.4 YUV 422 8-bit Mode

Data type is set to 0x1E.

Figure 42-8. YUV 422 8-bit Mode Bit

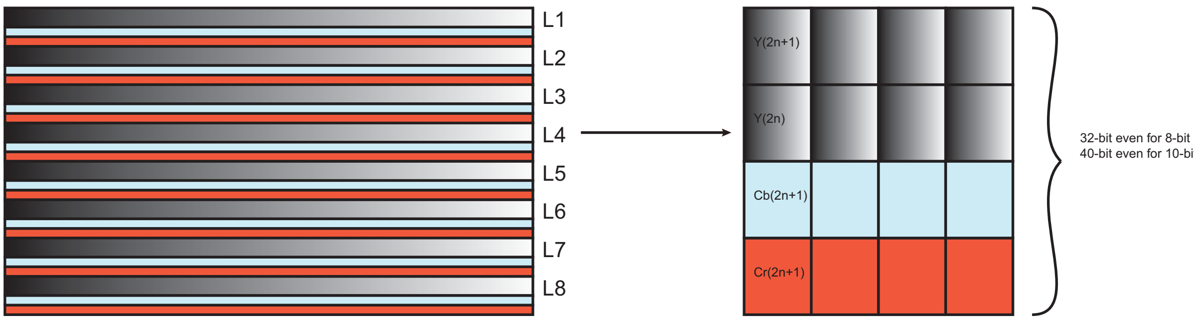


Table 42-20. YUV422 8-bit Mapping

DEMUX_DATA Slice	YUV 422 8-Bit Data Mapping
demux_data[39:30]	{Y2[7:0], 2'b00}
demux_data[29:20]	{Y1[7:0], 2'b00}
demux_data[19:10]	{U1[7:0], 2'b00} (Cb component}
demux_data[9:0]	{V1[7:0], 2'b00} (Cr component)

Table 42-21. YUV422 8-bit Recommended Memory Storage (Even Line)

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	U1								Y1								V1								Y2							
0x00	a7	a6	a5	a4	a3	a2	a1	a0	b7	b6	b5	b4	b3	b2	b1	b0	c7	v6	c5	c4	c3	c3	c1	c0	d7	d6	d5	d4	d3	d2	d1	d0
Pixel ID	U3								Y3								V3								Y4							
0x04	e7	e6	e5	e4	e3	e2	e1	e0	f7	f6	f5	f4	f3	f2	f1	f0	g7	g6	g5	g4	g3	g2	g1	g0	h7	h6	h5	h4	h3	h2	h1	h0

42.5.5.5 YUV 422 10-bit Mode

Data type is set to 0x1F.

Table 42-22. YUV422 8-bit Mapping

DEMUX_DATA Slice	YUV 422 8-Bit Data Mapping
demux_data[39:30]	Y2[9:0]
demux_data[29:20]	Y1[9:0]
demux_data[19:10]	U1[9:0] (Cb component}
demux_data[9:0]	V1[9:0] (Cr component)

Table 42-23. YUV422 10-bit Recommended Memory Storage

Addr	receive_buffer[31:24]	receive_buffer[23:16]	receive_buffer[15:8]	receive_buffer[7:0]
Pixel ID	Y2[9:2]	V1[9:2]	Y1[9:2]	U1[9:2]

...........continued

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
0x00	d9	d8	d7	d6	d5	d4	d3	d2	c9	c8	c7	c6	c5	c4	c3	c2	b9	b8	b7	b6	b5	b4	b3	b2	a9	a8	a7	a6	a5	a4	a3	a2
Pixel ID	V3[9:2]								Y3[9:2]								U3[9:2]								Y2[1:0]		V1[1:0]		Y1[1:0]		U1[1:0]	
0x04	g9	g8	g7	g6	g5	g4	g3	g2	f9	f8	f7	f6	f5	f4	f3	d2	e9	e8	e7	e6	e5	e4	e3	e2	d1	d0	c1	c0	b1	b0	a1	a0
Pixel ID	Y5[9:2]								U5[9:2]								Y4[1:0]		V3[1:0]		Y3[1:0]		U3[1:0]		Y4[9:2]							
0x08	j9	j8	j7	j6	j5	j4	j3	j2	i9	i8	i7	i6	i5	i4	i3	i2	h1	h0	g1	g0	f1	f0	e1	e0	h9	h8	h7	h6	h5	h4	h3	h2

42.5.6 RGB Packet Demux

Table 42-24. RGB Image Data Demultiplexing

Packet	Data Type	Handler
RGB image data	RGB888	Video pipe
	RGB666	Video pipe
	RGB565	Video pipe
	RGB555	Video pipe
	RGB444	Video pipe
	Reserved	RGB error interrupt

42.5.6.1 RGB888

Data type is set to 0x24.

Table 42-25. RGB888 Mapping for RGB36MAP=0

DEMUX_DATA Slice	RGB888 24-bit Data Mapping
demux_data[39:24]	0
demux_data[23:16]	R1[7:0]
demux_data[15:8]	G1[7:0]
demux_data[7:0]	B1[7:0]

Table 42-26. RGB888 Mapping for RGB36MAP=1

DEMUX_DATA Slice	RGB888 24-bit Data Mapping
demux_data[39:36]	0
demux_data[35:24]	R[7:0], R[7:4]
demux_data[23:12]	G[7:0], G[7:4]
demux_data[11:0]	B[7:0], B[7:4]

Table 42-27. RGB888 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	B2[7:0]								R1[7:0]								G1[7:0]								B1[7:0]							
0x00	d7	d6	d5	d4	d3	d2	d1	d0	c7	c6	c5	c4	c3	c2	c1	c0	b7	b6	b5	b4	b3	b2	b1	b0	a7	a6	a5	a4	a3	a2	a1	a0
Pixel ID	G3[7:0]								B3[7:0]								R2[7:0]								G2[7:0]							
0x04	h7	h6	h5	h4	h3	h2	h1	h0	g7	g6	g5	g4	g3	g2	g1	g0	f7	f6	f5	f4	f3	f2	f1	f0	e7	e6	e5	e4	e3	e2	e1	e0

42.5.6.2 RGB666

Data type is set to 0x23.

Table 42-28. RGB666 Mapping for RGB36MAP=0

DEMUX_DATA Slice	RGB666 18-bit Data Mapping
demux_data[39:18]	0
demux_data[17:12]	R1[5:0]
demux_data[11:6]	G1[5:0]
demux_data[5:0]	B1[5:0]

Table 42-29. RGB666 Mapping for RGB36MAP=1

DEMUX_DATA Slice	RGB666 18-bit Data Mapping
demux_data[39:36]	0
demux_data[35:24]	{R[5:0], R[5:0]}
demux_data[23:12]	{G[5:0], G[5:0]}
demux_data[11:0]	{B[5:0], B[5:0]}

Table 42-30. RGB666 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	R2		G2[5:0						B2[5:0]						R1[5:0]						G1[5:0]						B1[5:0]					
0x00	f1	f0	e5	e4	e3	e2	e1	e0	d5	d4	d3	d2	d1	d0	c5	c4	c3	c2	c1	c0	b5	b4	b3	b2	b1	b0	a5	a4	a3	a2	a1	a0
Pixel ID	G4				B4[5:0]						R3[5:0]						G3[5:0]						B3[5:0]						R2			
0x04	k3	k2	k1	k0	j5	j4	j3	j2	j1	j0	i5	i4	i3	i2	i1	i0	h5	h4	h3	h2	h1	h0	g5	g4	g3	g2	g1	g0	f5	f4	f3	f2
Pixel ID	B6[5:0]						R5[5:0]						G5[5:0]						B5[5:0]						R4[5:0]						G4	
0x08	p5	p4	p3	p2	p1	p0	o5	o4	o3	o2	o1	o0	n5	n4	n3	n2	n1	n0	m5	m4	m3	m2	m1	m0	l5	l4	l3	l2	l1	l0	k5	k4

42.5.6.3 RGB565

Data type is set to 0x22.

Table 42-31. RGB565 Mapping for RGB36MAP=0

DEMUX_DATA Slice	RGB565 16-bit Data Mapping
demux_data[39:16]	0
demux_data[15:11]	R1[4:0]
demux_data[10:5]	G1[5:0]
demux_data[4:0]	B1[4:0]

Table 42-32. RGB565 Mapping for RGB36MAP=1

DEMUX_DATA Slice	RGB565 16-bit Data Mapping
demux_data[39:36]	0
demux_data[35:24]	{R[4:0], R[4:0], R[4:3]}
demux_data[23:12]	{G[5:0], G[5:0]}
demux_data[11:0]	{B[4:0], B[4:0], B[4:3]}

42.5.6.4 RGB555

Data Type is set to 0x21.

Table 42-33. RGB555 Mapping for RGB36MAP=0

DEMUX_DATA Slice	RGB555 15-bit Data Mapping
demux_data[39:16]	0
demux_data[15:11]	R1[4:0]
demux_data[10:5]	{G1[4:0], 1’b0}
demux_data[4:0]	B1[4:0]

Table 42-34. RGB555 Mapping for RGB36MAP=1

DEMUX_DATA Slice	RGB555 15-bit Data Mapping
demux_data[39:36]	0
demux_data[35:24]	{R[4:0], R[4:0], R[4:3]}
demux_data[23:12]	{G[4:0], G[4:0], G[4:0]}
demux_data[11:0]	{B[4:0], B[4:0], B[4:3]}

42.5.6.5 RGB444

Data Type is set to 0x20.

Table 42-35. RGB444 Mapping for RGB36MAP=0

DEMUX_DATA Slice	RGB444 12-bit Data Mapping
demux_data[39:16]	0
demux_data[15:11]	R[3:0]
demux_data[10:5]	G[3:0]
demux_data[4:0]	B[3:0]

Table 42-36. RGB444 Mapping for RGB36MAP=1

DEMUX_DATA Slice	RGB444 12-bit Data Mapping
demux_data[39:36]	0
demux_data[35:24]	{R[3:0], R[3:0], R[3:0]}
demux_data[23:12]	{G[3:0], G[3:0], G[3:0]}
demux_data[11:0]	{B[3:0], B[3:0], B[3:0]}

42.5.7 RAW Packet Demux

Table 42-37. RAW Image Data Demultiplexing

Packet	Data Type	Handler
RAW image data	RAW6	Video pipe
	RAW7	Video pipe
	RAW8	Video pipe
	RAW10	Video pipe
	RAW12	Video pipe
	RAW14	Video pipe
	Reserved	RAW error interrupt

42.5.7.1 RAW6

Data type is set to 0x28.

Table 42-38. RAW6 Video Pipe Mapping

VPCFG.DE	VPCFG.PA	DEMUX_DATA Slice	RAW6 Data Mapping
0	0	demux_data[39:12]	0
		demux_data[11:0]	{6'b000000, RAW[5:0]}
0	1	demux_data[39:12]	0
		demux_data[11:0]	{RAW[5:0], 6'b000000}
1	1 (10-6-10)	demux_data[39:12]	0
		demux_data[39:12]	{RAW_DECODER[9:0], 2'b00}
1	1(12-6-12)	demux_data[39:12]	0
		demux_data[39:12]	RAW_DECODER[11:0]

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-39. RAW6 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P6		P5						P4						P3						P2						P1					
0x00	f1	f0	e5	e4	e3	e2	e1	e0	d5	d4	d3	d2	d1	d0	c5	c4	c3	c2	c1	c0	b5	b4	b3	b2	b1	b0	a5	a4	a3	a2	a1	a0
Pixel ID	P11				P10						P9						P8						P7						P6			
0x04	k3	k2	k1	k0	j5	j4	j3	j2	j1	j0	i5	i4	i3	i2	i1	i0	h5	h4	h3	h2	h1	h0	g5	g4	g3	g2	g1	g0	f5	f4	f3	f2

42.5.7.2 RAW7

Data type is set to 0x29.

Table 42-40. RAW7 Video Pipe Mapping

VPCFG.DE	VPCFG.PA	DEMUX_DATA Slice	RAW7 Data Mapping
0	0	demux_data[39:12]	0
		demux_data[11:0]	{5'b00000,RAW[6:0]}
0	1	demux_data[39:12]	0
		demux_data[11:0]	{RAW[6:0], 5'b00000}

...........continued

VPCFG.DE	VPCFG.PA	DEMUX_DATA Slice	RAW7 Data Mapping
1	1 (10-7-10)	demux_data[39:12]	0
		demux_data[11:0]	{RAW_DECODER[9:0],2’b00}
1	1 (12-7-12)	demux_data[39:12]	0
		demux_data[11:0]	RAW_DECODER[11:0]

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-41. RAW7 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P5				P4							P3							P2							P1						
0x00	e3	e2	e1	e0	d6	d5	d4	d3	d2	d1	d0	c6	c5	c4	c3	c2	c1	c0	b6	b5	b4	b3	b2	b1	b0	a6	a5	a4	a3	a2	a1	a0
Pixel ID		P9							P8							P7							P6							P5		
0x04	j0	i6	i5	i4	i3	i2	i1	i0	h6	h5	h4	h3	h2	h1	h0	g6	g5	g4	g3	g2	g1	g0	f6	f5	f4	f3	f2	f1	f0	e6	e5	e4

42.5.7.3 RAW8

Data type is set to 0x2A.

Table 42-42. RAW8 Mapping

VPCFG.DE	VPCFG.PA	DEMUX_DATA Slice	RAW8 Data Mapping
0	0	demux_data[39:12]	0
		demux_data[11:0]	{4’b0000, RAW[7:0]
0	1	demux_data[39:12]	0
		demux_data[11:0]	{RAW[7:0], 4’b0000}
1	1 (10-8-10)	demux_data[39:12]	0
		demux_data[11:0]	{RAW_DECODER[9:0],2’b00}
1	1(12-8-12)	demux_data[39:12]	0
		demux_data[11:0]	RAW_DECODER[11:0]

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-43. RAW8 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P4								P3								P2								P1							
0x00	d7	d6	d5	d4	d3	d2	d1	d0	c7	c6	c5	c4	c3	c2	c1	c0	b7	b6	b5	b4	b3	b2	b1	b0	a7	a6	a5	a4	a3	a2	a1	a0
Pixel ID	P8								P7								P6								P5							
0x04	h7	h6	h5	h4	h3	h2	h1	h0	g7	g6	g5	g4	g3	g2	g1	g0	f7	f6	f5	f4	f3	f2	f1	f0	e7	e6	e5	e4	e3	e2	e1	e0

42.5.7.4 RAW10

Data type is set to 0x2B.

Table 42-44. RAW10 Mapping

VPCFG.PA	DEMUX_DATA Slice	RAW10 Data Mapping
0	demux_data[39:12]	0
	demux_data[11:0]	{2’b00, RAW[9:0]}

...........continued		
VPCFG.PA	**DEMUX_DATA Slice**	**RAW10 Data Mapping**
1	demux_data[39:12]	0
	demux_data[11:0]	{RAW[9:0], 2’b00}

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-45. RAW10 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P4[9:2]								P3[9:2]								P2[9:2]								P1[9:2]							
0x00	d9	d8	d7	d6	d5	d4	d3	d2	c9	c8	c7	c6	c5	c4	c3	c2	b9	b8	b7	b6	b5	b4	b3	b2	a9	a8	a7	a6	a5	a4	a3	a2
Pixel ID	P7[9:2]								P6[9:2]								P5[9:2]								P4[1:0]		P3[1:0]		P2[1:0]		P1[1:0]	
0x04	g9	g8	g7	g6	g5	g4	g3	g2	f9	f8	f7	f6	f5	f4	f3	f2	e9	e8	e7	e6	e5	e4	e3	e2	d1	d0	c1	c0	b1	b0	a1	a0
Pixel ID	P10[9:2]								P9[9:2]								P8[1:0]		P7[1:0]		P6[1:0]		P5[1:0]		P8[9:2]							
0x08	j9	j8	j7	j6	j5	j4	j3	j2	i9	i8	i7	i6	i5	i4	i3	i2	h1	h0	g1	g0	f1	f0	e1	e0	h9	h8	h7	h6	h5	h4	h3	h2

42.5.7.5 RAW12

Data type is set to 0x2C.

Table 42-46. RAW12 Mapping

Single Configuration	DEMUX_DATA Slice	RAW10 Data Mapping
N/A	demux_data[39:12]	0
	demux_data[11:0]	RAW[11:0]

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-47. RAW12 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P3[11:4]								P2[3:0]				P1[3:0]				P2[11:4]								P1[11:4]							
0x00	c11	c10	c9	c8	c7	c6	c5	c4	b3	b2	b1	b0	a3	a2	a1	a0	b11	b10	b9	b8	b7	b6	b5	b4	a11	a10	a9	a8	a7	a6	a5	a4
Pixel ID	P6[11:4]								P5[11:4]								P4[3:0]				P3[3:0]				P4[11:4]							
0x04	f11	f10	f9	f8	f7	f6	f5	f4	e11	e10	e9	e8	e7	e6	e5	e4	d3	d2	d1	d0	c3	c2	c1	c0	d11	d10	d9	d8	d7	d6	d5	d4
Pixel ID	P8[3:0]				P7[3:0]				P8[11:4]								P7[11:4]								P6[3:0]				P5[3:0]			
0x08	h3	h2	h1	h0	g3	g2	g1	g0	h11	h10	h9	h8	h7	h6	h5	h4	g11	g10	g9	g8	g7	g6	g5	g4	f3	f2	f1	f0	e3	e2	e1	e0

42.5.7.6 RAW14

Data type is set to 0x2D.

Table 42-48. RAW14 Mapping

VPCFG.PA	DEMUX_DATA Slice	RAW10 Data Mapping
0	demux_data[39:14]	0
	demux_data[39:14]	RAW[13:0]
1	demux_data[39:12]	0
	demux_data[11:0]	RAW[13:2]

When the Recommended Memory Storage format is used, refer to the following table:

Table 42-49. RAW14 Recommended Memory Storage

Addr	receive_buffer[31:24]								receive_buffer[23:16]								receive_buffer[15:8]								receive_buffer[7:0]							
Pixel ID	P4[13:6]								P3[13:6]								P2[13:6]								P1[13:6]							
0x00	d13	d12	d11	d10	d9	d8	d7	d6	c13	c12	c11	c10	c9	c8	c7	c6	b13	b12	b11	b10	b9	b8	b7	b6	a13	a12	a11	a10	a9	a8	a7	a6
Pixel ID	P5[13:6]								P4[5:0]						P3[5:0]						P2[5:0]						P1[5:0]					
0x04	e13	e12	e11	e10	e9	e8	e7	e6	d5	d4	d3	d2	d1	d0	c5	c4	c3	c2	c1	c0	b5	b4	b3	b2	b1	b0	a5	a4	a3	a2	a1	a0
Pixel ID	P6[5:0]		P5[5:0]						P8[13:6]								P7[13:6]								P6[13:6]							
0x08	f1	f0	p5	p4	p3	p2	p1	p0	h13	h12	h11	h10	h9	h8	h7	h6	g13	g12	g11	g10	g9	g8	g7	g6	f13	f12	f11	f10	f9	f8	f7	f6
Pixel ID	P10[13:6]								P9[13:6]								P8[5:0]						P7[5:0]						P6[5:0]			
0x0C	q13	q12	q11	q10	q9	q8	q7	q6	p13	p12	p11	p10	p9	p8	p7	p6	h5	h4	h3	h2	h1	h0	g5	g4	g3	g2	g1	g0	f5	f4	f3	f2

42.5.8 User-Defined 8-bit Data

The User-Defined Data Type can be used to identify compressed data packets. CSI-2 specification indicates that a 12-7-12 compressed packet uses RAW7 data packing rule but the packet is tagged with User-Defined data type. The user-defined data type can be used to transmit arbitrary data such as JPEG or MPEG4.

Table 42-50. User-Defined 8-bit Data Demultiplexing

Packet	Data Type	Handler
User-defined 8-bit Data	User-defined 8-bit data type 1	Video pipe
	User-defined 8-bit data type 2	Video pipe
	User-defined 8-bit data type 3	Video pipe
	User-defined 8-bit data type 4	Video pipe
	User-defined 8-bit data type 5	Video pipe
	User-defined 8-bit data type 6	Video pipe
	User-defined 8-bit data type 7	Video pipe
	User-defined 8-bit data type 8	Video pipe

For user-defined data:

- The frame is transmitted as a sequence of arbitrarily-sized packets.
- The packet size may vary from packet to packet.
- Spacing between packets may vary.

42.5.9 CSI-2 Demux RAW Data Decompression Support

The CSI-2 implementation allows RAW data compression on the interface between the host processor and the camera module. Data compression schemes use an X-Y-Z naming convention where X is the number of bits per pixel in the original image, Y is the number of encoded (compressed) bits per pixel in the transmitted bit stream, and Z the number of decoded (uncompressed) bits per pixels.

Table 42-51. Decompression Mode Field Mapping to X-Y-Z Naming Convention

CSI2DC_VPCFG.DM	X-Y-Z CSI-2 Naming Rule
DECODER8TO12	12-8-12
DECODER7TO12	12-7-12
DECODER6TO12	12-6-12
DECODER8TO10	10-8-10
DECODER7TO10	10-7-10

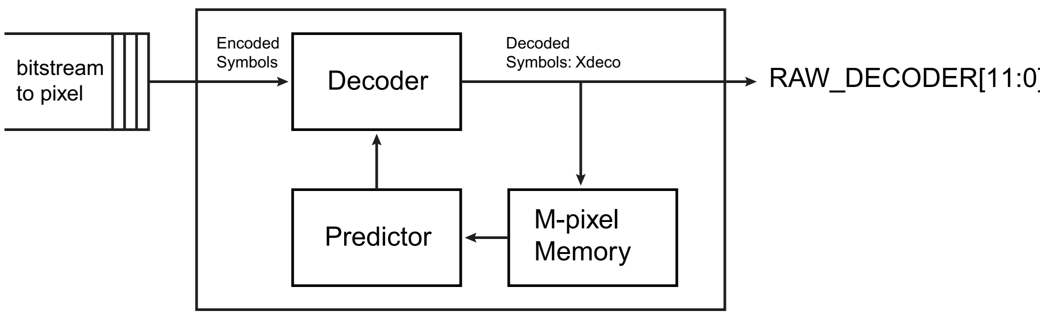
...........continued	
CSI2DC_VPCFG.DM	**X-Y-Z CSI-2 Naming Rule**
DECODER6TO10	10-6-10

The data compression schemes specified in CSI-2 Annex E are lossy and are designed to encode each line independently.

Table 42-52. Decoder Output Formatting

CSI2DC_VPCFG.DM	RAW_DECODER[11:0]
DECODER8TO12	XDECO[11:0]
DECODER7TO12	XDECO[11:0]
DECODER6TO12	XDECO[11:0]
DECODER8TO10	XDECO[9:0]
DECODER7TO10	XDECO[9:0]
DECODER6TO10	XDECO[9:0]

Figure 42-9. RAW Decompression Decoder



42.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CSI2DC_GCFGR	31:24								
		23:16								
		15:8								
		7:0	HLC[3:0]				SECDEDN	ULC	GPIOSEL	MIPIFRN
0x04	CSI2DC_GCTLR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x08	CSI2DC_GSR	31:24								
		23:16								
		15:8								
		7:0							ARSTIP	RSTIP
0x0C	CSI2DC_GIER	31:24								
		23:16								
		15:8								
		7:0	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
0x10	CSI2DC_GIDR	31:24								
		23:16								
		15:8								
		7:0	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
0x14	CSI2DC_GIMR	31:24								
		23:16								
		15:8								
		7:0	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
0x18	CSI2DC_GISR	31:24								
		23:16								
		15:8								
		7:0	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
0x1C	CSI2DC_SSPIER	31:24								
		23:16					RE[3:0]			
		15:8	LE[3:0]				LS[3:0]			
		7:0	FE[3:0]				FS[3:0]			
0x20	CSI2DC_SSPIDR	31:24								
		23:16					RE[3:0]			
		15:8	LE[3:0]				LS[3:0]			
		7:0	FE[3:0]				FS[3:0]			
0x24	CSI2DC_SSPIMR	31:24								
		23:16					RE[3:0]			
		15:8	LE[3:0]				LS[3:0]			
		7:0	FE[3:0]				FS[3:0]			
0x28	CSI2DC_SSPISR	31:24								
		23:16					RE[3:0]			
		15:8	LE[3:0]				LS[3:0]			
		7:0	FE[3:0]				FS[3:0]			
0x2C	CSI2DC_FNVC0R	31:24								
		23:16								
		15:8	FN[15:8]							
		7:0	FN[7:0]							
0x30	CSI2DC_FNVC1R	31:24								
		23:16								
		15:8	FN[15:8]							
		7:0	FN[7:0]							
0x34	CSI2DC_FNVC2R	31:24								
		23:16								
		15:8	FN[15:8]							
		7:0	FN[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	CSI2DC_FNVC3R	31:24								
		23:16								
		15:8	FN[15:8]							
		7:0	FN[7:0]							
0x3C	CSI2DC_LNVC0R	31:24								
		23:16								
		15:8	LN[15:8]							
		7:0	LN[7:0]							
0x40	CSI2DC_LNVC1R	31:24								
		23:16								
		15:8	LN[15:8]							
		7:0	LN[7:0]							
0x44	CSI2DC_LNVC2R	31:24								
		23:16								
		15:8	LN[15:8]							
		7:0	LN[7:0]							
0x48	CSI2DC_LNVC3R	31:24								
		23:16								
		15:8	LN[15:8]							
		7:0	LN[7:0]							
0x4C ... 0x5B	Reserved									
0x5C	CSI2DC_GSPIER	31:24								
		23:16								
		15:8								
		7:0	GSPERR[3:0]				GSPRDY[3:0]			
0x60	CSI2DC_GSPIDR	31:24								
		23:16								
		15:8								
		7:0	GSPERR[3:0]				GSPRDY[3:0]			
0x64	CSI2DC_GSPIMR	31:24								
		23:16								
		15:8								
		7:0	GSPERR[3:0]				GSPRDY[3:0]			
0x68	CSI2DC_GSPISR	31:24								
		23:16								
		15:8								
		7:0	GSPERR[3:0]				GSPRDY[3:0]			
0x6C	CSI2DC_GSPS0R	31:24								
		23:16			TYPE[5:0]					
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							
0x70	CSI2DC_GSPS1R	31:24								
		23:16			TYPE[5:0]					
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							
0x74	CSI2DC_GSPS2R	31:24								
		23:16			TYPE[5:0]					
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							
0x78	CSI2DC_GSPS3R	31:24								
		23:16			TYPE[5:0]					
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							
0x7C	CSI2DC_GLPIER	31:24								
		23:16								
		15:8	RE[3:0]				EB[3:0]			
		7:0	BL[3:0]				NU[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80	CSI2DC_GLPIDR	31:24								
		23:16								
		15:8	RE[3:0]				EB[3:0]			
		7:0	BL[3:0]				NU[3:0]			
0x84	CSI2DC_GLPIMR	31:24								
		23:16								
		15:8	RE[3:0]				EB[3:0]			
		7:0	BL[3:0]				NU[3:0]			
0x88	CSI2DC_GLPISR	31:24								
		23:16								
		15:8	RE[3:0]				EB[3:0]			
		7:0	BL[3:0]				NU[3:0]			
0x8C	CSI2DC_IDSCR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x90	CSI2DC_IDSIER	31:24								
		23:16								
		15:8								
		7:0				OVF	IDS[3:0]			
0x94	CSI2DC_IDSIDR	31:24								
		23:16								
		15:8								
		7:0				OVF	IDS[3:0]			
0x98	CSI2DC_IDSIMR	31:24								
		23:16								
		15:8								
		7:0				OVF	IDS[3:0]			
0x9C	CSI2DC_IDSISR	31:24								
		23:16								
		15:8								
		7:0				OVF	IDS[3:0]			
0xA0	CSI2DC_IDSEW0R0	31:24								
		23:16								
		15:8								
		7:0	VC[1:0]		DT[5:0]					
0xA4	CSI2DC_IDSEW1R0	31:24	RC[15:8]							
		23:16	RC[7:0]							
		15:8	WC[15:8]							
		7:0	WC[7:0]							
0xA8	CSI2DC_IDSEW0R1	31:24								
		23:16								
		15:8								
		7:0	VC[1:0]		DT[5:0]					
0xAC	CSI2DC_IDSEW1R1	31:24	RC[15:8]							
		23:16	RC[7:0]							
		15:8	WC[15:8]							
		7:0	WC[7:0]							
0xB0	CSI2DC_IDSEW0R2	31:24								
		23:16								
		15:8								
		7:0	VC[1:0]		DT[5:0]					
0xB4	CSI2DC_IDSEW1R2	31:24	RC[15:8]							
		23:16	RC[7:0]							
		15:8	WC[15:8]							
		7:0	WC[7:0]							

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0xB8	CSI2DC_IDSEW0R3	31:24								
		23:16								
		15:8								
		7:0	VC[1:0]		DT[5:0]					
0xBC	CSI2DC_IDSEW1R3	31:24	RC[15:8]							
		23:16	RC[7:0]							
		15:8	WC[15:8]							
		7:0	WC[7:0]							
0xC0	CSI2DC_PUR	31:24								
		23:16								
		15:8								
		7:0							DP	VP
0xC4	CSI2DC_PUSR	31:24	SIP							
		23:16								
		15:8								
		7:0							DP	VP
0xC8	CSI2DC_DPIER	31:24								
		23:16								
		15:8								
		7:0	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
0xCC	CSI2DC_DPIDR	31:24								
		23:16								
		15:8								
		7:0	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
0xD0	CSI2DC_DPIMR	31:24								
		23:16								
		15:8								
		7:0	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
0xD4	CSI2DC_DPISR	31:24								
		23:16								
		15:8								
		7:0	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
0xD8	CSI2DC_DPICR	31:24								
		23:16								
		15:8								
		7:0	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
0xDC	CSI2DC_DPER	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0xE0	CSI2DC_DPCFGR	31:24						BO[10:8]		
		23:16	BO[7:0]							
		15:8								
		7:0	VC[1:0]		DT[5:0]					
0xE4	CSI2DC_DPDCR	31:24	TC[15:8]							
		23:16	TC[7:0]							
		15:8								
		7:0		CSIZE[2:0]						DMA
0xE8	CSI2DC_VPIER	31:24								
		23:16								
		15:8								
		7:0			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
0xEC	CSI2DC_VPIDR	31:24								
		23:16								
		15:8								
		7:0			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xF0	CSI2DC_VPIMR	31:24								
		23:16								
		15:8								
		7:0			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
0xF4	CSI2DC_VPISR	31:24								
		23:16								
		15:8								
		7:0			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
0xF8	CSI2DC_VPER	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0xFC	CSI2DC_VPCFGR	31:24								
		23:16								
		15:8	RGB36MAP	PA	RMS	DP2	DM[2:0]			DE
		7:0	VC[1:0]		DT[5:0]					
0x0100	CSI2DC_VPCOLR	31:24								
		23:16								
		15:8	COL[15:8]							
		7:0	COL[7:0]							
0x0104	CSI2DC_VPROWR	31:24								
		23:16								
		15:8	ROW[15:8]							
		7:0	ROW[7:0]							
0x0108	CSI2DC_VPDTRR	31:24								
		23:16								
		15:8								
		7:0		DTRE	ADT[5:0]					

42.6.1 CSI2DC Global Configuration Register

Name: CSI2DC_GCFGR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	HLC[3:0]				SECDEDN	ULC	GPIOSEL	MIPIFRN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – HLC[3:0] CSI2DC Output Waveform Inter-line Minimum Delay
Inserts a minimal delay of (HLC+1) clock cycles between each line.

Bit 3 – SECDEDN Single Error Correction Double Error Detection Enable

Value	Description
0	Packet header error correction is activated.
1	Packet header error correction is disabled.

Bit 2 – ULC Use Optional Line Packet Delimiter

Value	Description
0	Line packets are not used to define the line boundary.
1	Line Start and Line End optional packets are used to activate and deactivate the line.

Bit 1 – GPIOSEL GPIO Parallel Interface Selection

Value	Description
0	The MIPI CSI-2 serial interface is activated.
1	The GPIO parallel interface is selected and internally routed to the Image Signal Processor.

Bit 0 – MIPIFRN MIPI Interface Free Running Clock

Value	Description
0	The sensor MIPI clock is free-running.
1	The sensor MIPI clock is gated.

42.6.2 CSI2DC Global Control Register

Name: CSI2DC_GCTLR
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Starts a software reset operation.

42.6.3 CSI2DC Global Status Register

Name: CSI2DC_GSR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ARSTIP	RSTIP
Access							R	R
Reset							0	0

Bit 1 – ARSTIP Asynchronous Reset in Progress
This bit can be cleared only if the D-PHY clock is running.

Value	Description
0	No reset in progress for the asynchronous domain.
1	Asynchronous domain is being reset.

Bit 0 – RSTIP Reset in Progress

Value	Description
0	No reset in progress for the synchronous domain.
1	Synchronous domain is being reset.

42.6.4 CSI2DC Global Interrupt Enable Register

Name: CSI2DC_GIER
Offset: 0x0C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – DED Packet Header Double Bit Error Detected Enable

Bit 6 – SEC Packet Header Single Bit Error Corrected Enable

Bit 5 – DP Data Pipe Interrupt Enable

Bit 4 – VP Video Pipe Interrupt Enable

Bit 3 – IDS Image Data Packet Snoop Controller Interrupt Enable

Bit 2 – GLP Generic Long Packet Interrupt Enable

Bit 1 – GSP Generic Short Packet Interrupt Enable

Bit 0 – SSP Synchronization Short Packet Interrupt Enable

42.6.5 CSI2DC Global Interrupt Disable Register

Name: CSI2DC_GIDR
Offset: 0x10
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – DED Double Bit Error Detected Interrupt Disable

Bit 6 – SEC Single Bit Error Corrected Interrupt Disable

Bit 5 – DP Data Pipe Interrupt Disable

Bit 4 – VP Video Pipe Interrupt Disable

Bit 3 – IDS Image Data Packet Snoop Controller Interrupt Disable

Bit 2 – GLP Generic Long Packet Interrupt Disable

Bit 1 – GSP Generic Short Packet Interrupt Disable

Bit 0 – SSP Synchronization Short Packet Interrupt Disable

42.6.6 CSI2DC Global Interrupt Mask Register

Name: CSI2DC_GIMR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is masked.

1: The corresponding interrupt is activated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – DED Double Error Detected Interrupt Disable Interrupt Mask

Bit 6 – SEC Single Error Corrected Interrupt Disable Interrupt Mask

Bit 5 – DP Data Pipe Interrupt Disable Interrupt Mask

Bit 4 – VP Video Pipe Interrupt Disable Interrupt Mask

Bit 3 – IDS Image Data Packet Snoop Controller Interrupt Mask

Bit 2 – GLP Generic Long Packet Interrupt Mask

Bit 1 – GSP Generic Short Packet Interrupt Mask

Bit 0 – SSP Synchronization Short Packet Interrupt Mask

42.6.7 CSI2DC Global Interrupt Status Register

Name: CSI2DC_GISR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DED	SEC	DP	VP	IDS	GLP	GSP	SSP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – DED Double Bit Error Detected Interrupt Status

Value	Description
0	No double bit error detected is pending.
1	A double bit error has been detected in the packet header. This bit is reset after the register read operation.

Bit 6 – SEC Single Bit Error Corrected Interrupt Status

Value	Description
0	No Single Bit Error Corrected interrupt is pending.
1	A single bit error has been detected and corrected in the packet header. This bit is reset after the register read operation.

Bit 5 – DP Data Pipe Interrupt Status

Value	Description
0	Either the interrupt source is masked at the DP level or no interrupt is pending for DP.
1	A Data Pipe interrupt is pending.

Bit 4 – VP Video Pipe Interrupt Status

Value	Description
0	Either the interrupt source is masked at the VP level or no interrupt is pending for VP.
1	A Video Pipe interrupt is pending.

Bit 3 – IDS Image Data Packet Snoop Controller Interrupt Status

Value	Description
0	Either the interrupt source is masked at the IDS level or no interrupt is pending for IDS.
1	A new Image Data Packet interrupt is pending.

Bit 2 – GLP Generic Long Packet Interrupt Status

Value	Description
0	Either the interrupt source is masked at the GLP level or no interrupt is pending for GLP.
1	A Generic Long Packet interrupt is pending.

Bit 1 – GSP Generic Short Packet Interrupt Status

Value	Description
0	Either the interrupt source is masked at the GSP level or no interrupt is pending for GSP.
1	A Generic Short Packet interrupt is pending.

Bit 0 – SSP Synchronization Short Packet Interrupt Status

Value	Description
0	Either the interrupt source is masked at the SSP level or no interrupt is pending for SSP.
1	A Synchronization Short Packet interrupt is pending.

42.6.8 CSI2DC SSP Interrupt Enable Register

Name: CSI2DC_SSPIER
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					RE[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	LE[3:0]				LS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	FE[3:0]				FS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 19:16 – RE[3:0] Reserved Short Packet Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the RE field will set the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Reserved Short Packet is detected.

Bits 15:12 – LE[3:0] Line End Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the LE field will set the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Line End is detected. Line Synchronization packets are optional.

Bits 11:8 – LS[3:0] Line Start Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the LS field will set the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Line Start is detected. Line Synchronization packets are optional.

Bits 7:4 – FE[3:0] Frame End Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the FE field will set the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Frame End is detected.

Bits 3:0 – FS[3:0] Frame Start Interrupt Enable

Value	Description
0	No effect.

Value	Description
1	Setting a bit at position i in the FS field will set the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Frame Start is detected.

42.6.9 CSI2DC SSP Interrupt Disable Register

Name: CSI2DC_SSPIDR
Offset: 0x20
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Setting a bit at position i in this field clears the interrupt mask bit for virtual channel i.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					RE[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	LE[3:0]				LS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	FE[3:0]				FS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 19:16 – RE[3:0] Reserved Short Packet Interrupt Disable

Bits 15:12 – LE[3:0] Line End Interrupt Disable

Bits 11:8 – LS[3:0] Line Start Interrupt Disable

Bits 7:4 – FE[3:0] Frame End Interrupt Disable

Bits 3:0 – FS[3:0] Frame Start Interrupt Disable

42.6.10 CSI2DC SSP Interrupt Mask Register

Name: CSI2DC_SSPIMR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					RE[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LE[3:0]				LS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FE[3:0]				FS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – RE[3:0] Reserved Short Packet Interrupt Mask

Value	Description
0	A bit cleared at position i in the field RE indicates that the Reserved Packet interrupt is masked for virtual channel i.
1	A bit set at position i in field RE indicates that Reserved Packet interrupt is activated for virtual channel i.

Bits 15:12 – LE[3:0] Line End Interrupt Mask

Value	Description
0	A bit cleared at position i in the field LE indicates that the Line End interrupt is masked for virtual channel i.
1	A bit set at position i in field LE indicates that Line End interrupt is activated for virtual channel i.

Bits 11:8 – LS[3:0] Line Start Interrupt Mask

Value	Description
0	A bit cleared at position i in the field LS indicates that the Line Start interrupt is masked for virtual channel i.
1	A bit set at position i in field LS indicates that Line Start interrupt is activated for virtual channel i.

Bits 7:4 – FE[3:0] Frame End Interrupt Mask

Value	Description
0	A bit cleared at position i in the field FE indicates that the Frame End interrupt is masked for virtual channel i.
1	A bit set at position i in field FE indicates that Frame End interrupt is activated for virtual channel i.

Bits 3:0 – FS[3:0] Frame Start Interrupt Mask

Value	Description
0	A bit cleared at position i in the field FS indicates that the Frame Start interrupt is masked for virtual channel i.

Value	Description
1	A bit set at position i in field FS indicates that Frame Start interrupt is activated for virtual channel i.

42.6.11 CSI2DC SSP Interrupt Status Register

Name: CSI2DC_SSPISR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					RE[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LE[3:0]				LS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FE[3:0]				FS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – RE[3:0] Reserved Short Packet Interrupt Mask

Value	Description
0	A bit cleared at position i in the field RE indicates that no Reserved Short Packet interrupt is pending for virtual channel i.
1	A bit set at position i in the field RE indicates that no Reserved Short Packet interrupt is pending for virtual channel i.

Bits 15:12 – LE[3:0] Line End Interrupt Mask

Value	Description
0	A bit cleared at position i in the field LE indicates that no Line End interrupt is pending for virtual channel i.
1	A bit set at position i in the field LE indicates that a Line End interrupt is pending for virtual channel i.

Bits 11:8 – LS[3:0] Line Start Interrupt Mask

Value	Description
0	A bit cleared at position i in the field LS indicates that no Line Start interrupt is pending for virtual channel i.
1	A bit set at position i in the field LS indicates that a Line Start interrupt is pending for virtual channel i.

Bits 7:4 – FE[3:0] Frame End Interrupt Mask

Value	Description
0	A bit cleared at position i in the field FE indicates that no Frame End interrupt is pending for virtual channel i.
1	A bit set at position i in the field FE indicates that a Frame End interrupt is pending for virtual channel i.

Bits 3:0 – FS[3:0] Frame Start Interrupt Mask

Value	Description
0	A bit cleared at position i in the field FS indicates that no Frame Start interrupt is pending for virtual channel i.

Value	Description
1	A bit set at position i in the field FS indicates that a Frame Start interrupt is pending for virtual channel i.

42.6.12 CSI2DC Frame Number Virtual Channel 0 Register

Name: CSI2DC_FNVC0R
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FN[15:0] Frame Number for Virtual Channel 0

42.6.13 CSI2DC Frame Number Virtual Channel 1 Register

Name: CSI2DC_FNVC1R
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FN[15:0] Frame Number for Virtual Channel 1

42.6.14 CSI2DC Frame Number Virtual Channel 2 Register

Name: CSI2DC_FNVC2R
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	FN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FN[15:0] Frame Number for Virtual Channel 2

42.6.15 CSI2DC Frame Number Virtual Channel 3 Register

Name: CSI2DC_FNVC3R
Offset: 0x38
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FN[15:0] Frame Number for Virtual Channel 3

42.6.16 CSI2DC Line Number Virtual Channel 0 Register

Name: CSI2DC_LNVC0R
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – LN[15:0] Line Number for Virtual Channel 0

42.6.17 CSI2DC Line Number Virtual Channel 1 Register

Name: CSI2DC_LNVC1R
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – LN[15:0] Line Number for Virtual Channel 1

42.6.18 CSI2DC Line Number Virtual Channel 2 Register

Name: CSI2DC_LNVC2R
Offset: 0x44
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – LN[15:0] Line Number for Virtual Channel 2

42.6.19 CSI2DC Line Number Virtual Channel 3 Register

Name: CSI2DC_LNVC3R
Offset: 0x48
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – LN[15:0] Line Number for Virtual Channel 3

42.6.20 CSI2DC GSP Interrupt Enable Register

Name: CSI2DC_GSPIER
Offset: 0x5C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	GSPERR[3:0]				GSPRDY[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:4 – GSPERR[3:0] Generic Short Packet Error Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the GSPERR field will set the interrupt mask bit, and virtual channel i can generate a Generic Short Packet Error interrupt.

Bits 3:0 – GSPRDY[3:0] Generic Short Packet Ready Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the GSPRDY field will set the interrupt mask bit, and virtual channel i can generate a Generic Short Packet interrupt.

42.6.21 CSI2DC GSP Interrupt Disable Register

Name: CSI2DC_GSPIDR
Offset: 0x60
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	GSPERR[3:0]				GSPRDY[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:4 – GSPERR[3:0] Generic Short Packet Error Interrupt Disable

Value	Description
0	No effect.
1	Setting a bit at position i in the GSPERR field will clear the interrupt mask bit for virtual channel i.

Bits 3:0 – GSPRDY[3:0] Generic Short Packet Ready Interrupt Disable

Value	Description
0	No effect.
1	Setting a bit at position i in the GSPRDY field will clear the interrupt mask bit for virtual channel i.

42.6.22 CSI2DC GSP Interrupt Mask Register

Name: CSI2DC_GSPIMR
Offset: 0x64
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	GSPERR[3:0]				GSPRDY[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – GSPERR[3:0] Generic Short Packet Error Interrupt Mask bit

Value	Description
0	A bit cleared at position i in the field GSPERR indicates that the GSP Error interrupt is masked.
1	A bit set at position i in field GSPERR indicates that the GSP Error interrupt is activated.

Bits 3:0 – GSPRDY[3:0] Generic Short Packet Ready Interrupt Mask bit

Value	Description
0	A bit cleared at position i in the field GSPRDY indicates that the GSP Ready interrupt is masked for virtual channel i.
1	A bit set at position i in field GSPRDY indicates that the GSP Ready interrupt is activated for virtual channel i.

42.6.23 CSI2DC GSP Interrupt Status Register

Name: CSI2DC_GSPISR
Offset: 0x68
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	GSPERR[3:0]				GSPRDY[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – GSPERR[3:0] Generic Short Packet Error Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field GSPERR indicates that no Generic Short Packet Error interrupt is pending for virtual channel i.
1	A bit set at position i in the field GSPERR indicates that a Generic Short Packet Error overflow interrupt has occurred since the last read of the status register.

Bits 3:0 – GSPRDY[3:0] Generic Short Packet Ready Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field GSPRDY indicates that no Generic Short Packet Ready interrupt is pending for virtual channel i.
1	A bit set at position i in the field GSPRDY indicates that a Generic Short Packet Ready interrupt is pending for virtual channel i.

42.6.24 CSI2DC GSP Status 0 Register

Name: CSI2DC_GSPS0R
Offset: 0x6C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TYPE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 21:16 – TYPE[5:0] Generic Short Packet Type Value for Virtual Channel 0

Bits 15:0 – VALUE[15:0] Generic Short Packet 16-bit Data Value for Virtual Channel 0
This field is a data value that must be transmitted to the application layer.

42.6.25 CSI2DC GSP Status 1 Register

Name: CSI2DC_GSPS1R
Offset: 0x70
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			TYPE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 21:16 – TYPE[5:0] Generic Short Packet Type Value for Virtual Channel 1

Bits 15:0 – VALUE[15:0] Generic Short Packet 16-bit Data Value for Virtual Channel 1
This field is a data value that must be transmitted to the application layer.

42.6.26 CSI2DC GSP Status 2 Register

Name: CSI2DC_GSPS2R
Offset: 0x74
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TYPE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 21:16 – TYPE[5:0] Generic Short Packet Type Value for Virtual Channel 2

Bits 15:0 – VALUE[15:0] Generic Short Packet 16-bit Data Value for Virtual Channel 2
This field is a data value that must be transmitted to the application layer.

42.6.27 CSI2DC GSP Status 3 Register

Name: CSI2DC_GSPS3R
Offset: 0x78
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TYPE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 21:16 – TYPE[5:0] Generic Short Packet Type Value for Virtual Channel 3

Bits 15:0 – VALUE[15:0] Generic Short Packet 16-bit Data Value for Virtual Channel 3
This field is a data value that must be transmitted to the application layer.

42.6.28 CSI2DC GLP Interrupt Enable Register

Name: CSI2DC_GLPIER
Offset: 0x7C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RE[3:0]				EB[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	BL[3:0]				NU[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 15:12 – RE[3:0] Reserved Packet Interrupt Enable Bit

Value	Description
0	No effect.
1	Setting a bit at position i in the RE field sets the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a reserved long packet is detected. A reserved long packet includes data types from 0x13 to 0x17.

Bits 11:8 – EB[3:0] Embedded 8-bit Non-Image Data Interrupt Enable Bit

Value	Description
0	No effect.
1	Setting a bit at position i in the EB field sets the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a non-image data packet is detected.

Bits 7:4 – BL[3:0] Blanking Data Interrupt Enable Bit

Value	Description
0	No effect.
1	Setting a bit at position i in the BL field sets the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a Blank packet is detected.

Bits 3:0 – NU[3:0] Null Interrupt Enable Bit

Value	Description
0	No effect.
1	Setting a bit at position i in the NU field sets the interrupt mask bit for virtual channel i, and this virtual channel can generate an interrupt when a null long packet is detected.

42.6.29 CSI2DC GLP Interrupt Disable Register

Name: CSI2DC_GLPIDR
Offset: 0x80
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Setting a bit at position i in this field clears the interrupt mask bit for Virtual Channel i.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RE[3:0]				EB[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	BL[3:0]				NU[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 15:12 – RE[3:0] Reserved Packet Interrupt Disable Bit

Bits 11:8 – EB[3:0] Embedded 8-bit Non-Image Data Interrupt Disable Bit

Bits 7:4 – BL[3:0] Blanking Data Interrupt Disable Bit

Bits 3:0 – NU[3:0] Null Interrupt Disable Bit

42.6.30 CSI2DC GLP Interrupt Mask Register

Name: CSI2DC_GLPIMR
Offset: 0x84
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RE[3:0]				EB[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BL[3:0]				NU[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – RE[3:0] Reserved Packet Interrupt Mask Bit

Value	Description
0	A bit cleared at position i in the field RE indicates that the Reserved Packet interrupt is masked for virtual channel i.
1	A bit set at position i in field NU indicates that the Reserved Packet interrupt is activated for virtual channel i.

Bits 11:8 – EB[3:0] Embedded 8-bit Non-Image Data Packet Interrupt Mask Bit

Value	Description
0	A bit cleared at position i in the field EB indicates that the embedded data packet interrupt is masked for virtual channel i.
1	A bit set at position i in field EB indicates that the embedded data packet interrupt is activated for virtual channel i.

Bits 7:4 – BL[3:0] Blanking Data Packet Interrupt Mask Bit

Value	Description
0	A bit cleared at position i in the field BL indicates that the Blanking Data Packet interrupt is masked for virtual channel i.
1	A bit set at position i in field BL indicates that the Blanking Data Packet interrupt is activated for virtual channel i.

Bits 3:0 – NU[3:0] Null Packet Interrupt Mask Bit

Value	Description
0	A bit cleared at position i in the field NU indicates that the Null Packet interrupt is masked for virtual channel i.
1	A bit set at position i in field NU indicates that the Null Packet interrupt is activated for virtual channel i.

42.6.31 CSI2DC GLP Interrupt Status Register

Name: CSI2DC_GLPISR
Offset: 0x88
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RE[3:0]				EB[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BL[3:0]				NU[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – RE[3:0] Reserved Generic Long Packet Ready Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field BL indicates that no reserved packet interrupt is pending for virtual channel i.
1	A bit set at position i in the field BL indicates that a reserved packet interrupt is pending for virtual channel i.

Bits 11:8 – EB[3:0] Embedded 8-bit data Generic Long Packet Ready Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field EB indicates that no embedded data packet interrupt is pending for virtual channel i.
1	A bit set at position i in the field EB indicates that an embedded data packet interrupt is pending for virtual channel i.

Bits 7:4 – BL[3:0] Blanking Data Generic Long Packet Ready Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field BL indicates that no blanking data packet interrupt is pending for virtual channel i.
1	A bit set at position i in the field BL indicates that a blanking packet interrupt is pending for virtual channel i.

Bits 3:0 – NU[3:0] Null Generic Long Packet Ready Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field NU indicates that no null packet interrupt is pending for virtual channel i.
1	A bit set at position i in the field NU indicates that a null packet interrupt is pending for virtual channel i.

42.6.32 CSI2DC IDS Control Register

Name: CSI2DC_IDSCR
Offset: 0x8C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Performs an IDS software reset of the table. Read value when set indicates that the software reset is in progress.

42.6.33 CSI2DC IDS Interrupt Enable Register

Name: CSI2DC_IDSIER
Offset: 0x90
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				OVF	IDS[3:0]			
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 4 – OVF Image Data Snoop Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the Image Data Snoop Overflow interrupt.

Bits 3:0 – IDS[3:0] Image Data Snoop Interrupt Enable

Value	Description
0	No effect.
1	Setting a bit at position i in the IDS field will set the interrupt mask bit for table entry i, and this entry can generate an interrupt when an image data packet is captured by the snoop controller.

42.6.34 CSI2DC IDS Interrupt Disable Register

Name: CSI2DC_IDSIDR
Offset: 0x94
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				OVF	IDS[3:0]			
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 4 – OVF Image Data Snoop Overflow Interrupt Disable

Value	Description
0	No effect.
1	Disables the Image Data Snoop Overflow interrupt.

Bits 3:0 – IDS[3:0] Image Data Snoop Interrupt Disable

Value	Description
0	No effect.
1	Setting a bit at position i in the IDS field will clear the interrupt mask bit for table entry i.

42.6.35 CSI2DC IDS Interrupt Mask Register

Name: CSI2DC_IDSIMR
Offset: 0x98
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				OVF	IDS[3:0]			
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – OVF Image Data Snoop Overflow Interrupt Mask

Value	Description
0	No effect.
1	Indicates that the Image Data Snoop Overflow interrupt is activated.

Bits 3:0 – IDS[3:0] Image Data Snoop Interrupt Mask Bit

Value	Description
0	A bit cleared at position i in the field IDS indicates that the Image Data Snoop interrupt is masked for table entry i.
1	A bit set at position i in field IDS indicates that Image Data Snoop interrupt is activated for table entry i.

42.6.36 CSI2DC IDS Interrupt Status Register

Name: CSI2DC_IDSISR
Offset: 0x9C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				OVF	IDS[3:0]			
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – OVF Image Data Snoop Overflow Interrupt Status

Value	Description
0	No effect.
1	Indicates that the IDS controller captured more than 4 entries.

Bits 3:0 – IDS[3:0] Image Data Snoop Interrupt Status Bit

Value	Description
0	A bit cleared at position i in the field IDS indicates that no Image Data Snoop interrupt is pending for virtual channel i.
1	A bit set at position i in the field IDS indicates that a new Image Data Snoop entry interrupt is pending for table entry i.

42.6.37 CSI2DC IDS Entry Word 0 Register

Name: CSI2DC_IDSEW0Rx
Offset: 0xA0 + x*0x08 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	VC[1:0]		DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:6 – VC[1:0] Virtual Channel Identifier

Value	Name	Description
0	VC0	Virtual Channel 0
1	VC1	Virtual Channel 1
2	VC2	Virtual Channel 2
3	VC3	Virtual Channel 3

Bits 5:0 – DT[5:0] Data Type
Indicates the value of the data type for the captured packet.

42.6.38 CSI2DC IDS Entry Word 1 Register

Name: CSI2DC_IDSEW1Rx
Offset: 0xA4 + x*0x08 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – RC[15:0] Row Count for Image Data Packet Captured
Row count for the current packet. The number of rows in the image is RC+1.

Bits 15:0 – WC[15:0] Word Count for Image Data Packet Captured
Packet word count.

42.6.39 CSI2DC Pipe Update Register

Name: CSI2DC_PUR
Offset: 0xC0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DP	VP
Access							W	W
Reset							–	–

Bit 1 – DP Data Pipe Attributes Update

Value	Description
0	No effect.
1	Transfers current configuration to Data Pipe Configuration registers on the next Frame Start Packet detection if the FS packet virtual channel ID matches the CSI2DC_DPCFG.VC field. This field must be set after data pipe configuration, otherwise the settings will not be updated.

Bit 0 – VP Video Pipe Attributes Update

Value	Description
0	No effect.
1	Transfers current configuration to Video Pipe Configuration registers on the next Frame Start Packet detection if the FS packet virtual channel ID matches the CSI2DC_VPCFG.VC field. This field must be set after video pipe configuration, otherwise the settings will not be updated.

42.6.40 CSI2DC Pipe Update Status Register

Name: CSI2DC_PUSR
Offset: 0xC4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SIP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DP	VP
Access							R	R
Reset							0	0

Bit 31 – SIP Synchronization In Progress

Value	Description
0	No synchronization pending.
1	Synchronization across clock domain boundary is in progress. If the MIPI interface clock is gated, the synchronization procedure will wait for the first valid MIPI packet to activate the receiver clock.

Bit 1 – DP Data Pipe Update

Value	Description
0	No data pipe in progress.
1	Data pipe configuration is in progress. This bit is cleared at the next frame start packet if the virtual channel identifier matches the CSI2DC_DPCFG.VC field.

Bit 0 – VP Video Pipe Update

Value	Description
0	No video pipe in progress.
1	Video pipe configuration is in progress. This bit is cleared at the next frame start packet if the virtual channel identifier matches the CSI2DC_VPCFG.VC field.

42.6.41 CSI2DC Data Pipe Interrupt Enable Register

Name: CSI2DC_DPIER
Offset: 0xC8
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – LTE Longer Than Expected Packet Received Interrupt Enable Bit

Bit 6 – STE Shorter Than Expected Packet Received Interrupt Enable

Bit 5 – DATOVF Data Pipe Overflow Interrupt Enable

Bit 4 – RXOVF1 Bank 1, Packet Overflow Interrupt Enable

Bit 3 – RXOVF0 Bank 0, Packet Overflow Interrupt Enable

Bit 2 – RXRDY1 Bank 1, Packet Received Interrupt Enable

Bit 1 – RXRDY0 Bank 0, Packet Received Interrupt Enable

Bit 0 – CAPTURE Data Pipe Capture Done Interrupt Enable

42.6.42 CSI2DC Data Pipe Interrupt Disable Register

Name: CSI2DC_DPIDR
Offset: 0xCC
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – LTE Longer Than Expected Packet Received Interrupt Disable

Bit 6 – STE Shorter Than Expected Packet Received Interrupt Disable

Bit 5 – DATOVF Data Pipe Overflow Interrupt Disable

Bit 4 – RXOVF1 Bank 1, Packet Overflow Interrupt Disable

Bit 3 – RXOVF0 Bank 0, Packet Overflow Interrupt Disable

Bit 2 – RXRDY1 Bank 1, Packet Received Interrupt Disable

Bit 1 – RXRDY0 Bank 0, Packet Received Interrupt Disable

Bit 0 – CAPTURE Data Pipe Capture Done Interrupt Disable

42.6.43 CSI2DC Data Pipe Interrupt Mask Register

Name: CSI2DC_DPIMR
Offset: 0xD0
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – LTE Longer Than Expected Packet Received Interrupt Mask

Bit 6 – STE Shorter Than Expected Packet Received Interrupt Mask

Bit 5 – DATOVF Data Pipe Overflow Interrupt Mask

Bit 4 – RXOVF1 Bank 1, Packet Overflow Interrupt Mask

Bit 3 – RXOVF0 Bank 0, Packet Overflow Interrupt Mask

Bit 2 – RXRDY1 Bank 1, Packet Received Interrupt Mask

Bit 1 – RXRDY0 Bank 0, Packet Received Interrupt Mask

Bit 0 – CAPTURE Data Pipe Capture Done Interrupt Mask

42.6.44 CSI2DC Data Pipe Interrupt Status Register

Name: CSI2DC_DPISR
Offset: 0xD4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – LTE Packet Longer Than Expected

Value	Description
0	No LTE packet detected.
1	A packet has been received but the actual length is longer that the packet word count value.

Bit 6 – STE Packet Shorter Than Expected

Value	Description
0	No STE packet detected.
1	A packet has been received but the actual length is shorter that the packet word count value.

Bit 5 – DATOVF Data Overflow

Value	Description
0	No overflow detected.
1	Data overflow in the clock domain crossing FIFO.

Bit 4 – RXOVF1 Bank 1 Overflow

Value	Description
0	No overflow detected.
1	An overflow occurred in bank 1.

Bit 3 – RXOVF0 Bank 0 Overflow

Value	Description
0	No overflow detected.
1	An overflow occurred in bank 0.

Bit 2 – RXRDY1 Bank 1 Packet Received

Value	Description
0	No packet received in bank 1 since the last clear operation of the register.
1	A new packet has been captured in the data pipe.

Bit 1 – RXRDY0 Bank 0 Packet Received

Value	Description
0	No packet received in bank 0 since the last clear operation of the register.
1	A new packet has been captured in the data pipe.

Bit 0 – CAPTURE Captured Frame

Value	Description
0	No frame captured on the data pipe interface since the last clear operation of the register.
1	A new frame has been captured in the data pipe.

42.6.45 CSI2DC Data Pipe Interrupt Clear Register

Name: CSI2DC_DPICR
Offset: 0xD8
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LTE	STE	DATOVF	RXOVF1	RXOVF0	RXRDY1	RXRDY0	CAPTURE
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – LTE Packet Longer Than Expected Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the LTE interrupt.

Bit 6 – STE Packet Shorter Than Expected Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the STE interrupt.

Bit 5 – DATOVF Data Overflow Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Data Overflow interrupt.

Bit 4 – RXOVF1 Bank 1 Packet Overflow Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Bank 1 Packet Overflow interrupt.

Bit 3 – RXOVF0 Bank 0 Packet Overflow Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Bank 0 Packet Overflow interrupt.

Bit 2 – RXRDY1 Bank 1 Packet Received Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Bank 1 Packet Received interrupt.

Bit 1 – RXRDY0 Bank 0 Packet Received Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Bank 0 Packet Received interrupt.

Bit 0 – CAPTURE Captured Frame Interrupt Clear Register

Value	Description
0	No effect.
1	Clears the Captured Frame interrupt.

42.6.46 CSI2DC Data Pipe Enable Register

Name: CSI2DC_DPER
Offset: 0xDC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Data Pipe Enable

Value	Description
0	Data pipe disabled.
1	Data pipe enabled.

42.6.47 CSI2DC Data Pipe Configuration Register

Name: CSI2DC_DPCFGR
Offset: 0xE0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						BO[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	BO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	VC[1:0]		DT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – BO[10:0] Bank Offset
When a packet is received in the data pipe, this field indicates the starting memory location of bank 1. Each memory location is 32 bits.

Bits 7:6 – VC[1:0] Virtual Channel for Data Pipe

Bits 5:0 – DT[5:0] Data Type for Data Pipe
Indicates the data type for the data pipe. The DT must match the desired packet content.

42.6.48 CSI2DC Data Pipe DMA Configuration Register

Name: CSI2DC_DPDCR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CSIZE[2:0]						DMA
Access		R/W	R/W	R/W				R/W
Reset		0	0	0				0

Bits 31:16 – TC[15:0] DMA Transfer Count
Indicates the number of data to be transferred.

Bits 6:4 – CSIZE[2:0] DMA Chunk Size

Value	Name	Description
0	CHK_1	1 data transferred
1	CHK_2	2 data transferred
2	CHK_4	4 data transferred
3	CHK_8	8 data transferred
4	CHK_16	16 data transferred

Bit 0 – DMA DMA Mode Enabled

Value	Description
0	DMA client interface is disabled.
1	DMA client interface is enabled.

42.6.49 CSI2DC Video Pipe Interrupt Enable Register

Name: CSI2DC_VPIER
Offset: 0xE8
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – PKTOVF Packet Overflow For Video Pipe Interrupt Enable

Bit 4 – LTE Packet Longer Than Expected Interrupt Enable

Bit 3 – STE Packet Shorter Than Expected Interrupt Enable

Bit 2 – CTLOVF Control Buffer Overflow Interrupt Enable

Bit 1 – RATEOVF Rate Buffer Overflow Interrupt Enable

Bit 0 – CAPTURE Video Pipeline Capture Interrupt Enable

42.6.50 CSI2DC Video Pipe Interrupt Disable Register

Name: CSI2DC_VPIDR
Offset: 0xEC
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – PKTOVF Packet Overflow For Video Pipe Interrupt Disable

Bit 4 – LTE Packet Longer Than Expected Interrupt Disable

Bit 3 – STE Packet Shorter Than Expected Interrupt Disable

Bit 2 – CTLOVF Control Buffer Overflow Interrupt Disable

Bit 1 – RATEOVF Rate Buffer Overflow Interrupt Disable

Bit 0 – CAPTURE Video Pipeline Capture Interrupt Disable

42.6.51 CSI2DC Video Pipe Interrupt Mask Register

Name: CSI2DC_VPIMR
Offset: 0xF0
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is masked.

1: The corresponding interrupt is activated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – PKTOVF Packet Overflow For Video Pipe Interrupt Mask

Bit 4 – LTE Packet Longer Than Expected Interrupt Mask

Bit 3 – STE Packet Shorter Than Expected Interrupt Mask

Bit 2 – CTLOVF Control Buffer Overflow Interrupt Mask

Bit 1 – RATEOVF Rate Buffer Overflow Interrupt Mask

Bit 0 – CAPTURE Video Pipeline Capture Interrupt Mask

42.6.52 CSI2DC Video Pipe Interrupt Status Register

Name: CSI2DC_VPISR
Offset: 0xF4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PKTOVF	LTE	STE	CTLOVF	RATEOVF	CAPTURE
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – PKTOVF Packet Overflow For Video Pipe Interrupt Mask

Value	Description
0	Packet Overflow For Video Pipe interrupt is masked.
1	Packet Overflow For Video Pipe interrupt is activated.

Bit 4 – LTE Packet Longer Than Expected Interrupt Mask

Value	Description
0	Packet Longer Than Expected interrupt is masked.
1	Packet Longer Than Expected interrupt is activated.

Bit 3 – STE Packet Shorter Than Expected Interrupt Mask

Value	Description
0	No packet shorter than expected since the last read of the register.
1	The Packet Shorter Than Expected interrupt is activated.

Bit 2 – CTLOVF Control Buffer Overflow Interrupt Mask

Value	Description
0	No Control Buffer Overflow since the last read of the register.
1	A Control Buffer Overflow has been detected.

Bit 1 – RATEOVF Rate Buffer Overflow Interrupt Mask

Value	Description
0	No Rate Buffer Overflow since the last read of the register
1	A Rate Buffer Overflow has been detected.

Bit 0 – CAPTURE Video Pipeline Capture Status

Value	Description
0	No frame capture since the last read of the register.
1	A frame has been captured in the video pipeline.

42.6.53 CSI2DC Video Pipe Enable Register

Name: CSI2DC_VPER
Offset: 0xF8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Video Pipe Enable

Value	Description
0	Video pipe disabled.
1	Video pipe enabled.

42.6.54 CSI2DC Video Pipe Configuration Register

Name: CSI2DC_VPCFGR
Offset: 0xFC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RGB36MAP	PA	RMS	DP2		DM[2:0]		DE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VC[1:0]				DT[5:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – RGB36MAP RGB Mapping

Value	Description
0	RGB data is packed and written to the output bus.
1	The RGB pixel is mapped onto the 36-bit bus, using pixel expansion and replication.

Bit 14 – PA ISC Post Adjustment

Value	Description
0	Post adjustment is disabled. Video pipe output data are LSB aligned (left untouched).
1	Post adjustment is enabled. Video pipe output data are MSB aligned according to 12-bit ISC data bus.

Bit 13 – RMS Recommended Memory Storage

Value	Description
0	CSI2DC outputs 1 pixel per component per clock cycle, compliant with the ISC processing engine.
1	CSI2DC generates a byte stream compliant with the CSI-2 specification memory format.

Bit 12 – DP2 Decoder Predictor 2 Selection

Value	Description
0	Predictor 1 is selected.
1	Predictor 2 is selected.

Bits 11:9 – DM[2:0] Decoder Mode
See CSI-2 Demux RAW Data Decompression Support.

Value	Name	Description
0	DECODER8TO12	Use the 8-bit to 12-bit decoding operation
1	DECODER7TO12	Use the 7-bit to 12-bit decoding operation
2	DECODER6TO12	Use the 6-bit to 12-bit decoding operation
3	DECODER8TO10	Use the 8-bit to 10-bit decoding operation
4	DECODER7TO10	Use the 7-bit to 10-bit decoding operation

Value	Name	Description
5	DECODER6TO10	Use the 6-bit to 10-bit decoding operation

Bit 8 – DE Decompression Enable

Value	Description
0	Decompression disabled.
1	Decompression enabled.

Bits 7:6 – VC[1:0] Virtual Channel Identifier
Must be configured with the camera virtual channel identifier.

Bits 5:0 – DT[5:0] Data Type
Must be configured with the desired image data type.

42.6.55 CSI2DC Video Pipe Column Register

Name: CSI2DC_VPCOLR
Offset: 0x100
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COL[15:0] Column Number
Current active column number of the frame being processed in the video pipe

42.6.56 CSI2DC Video Pipe Row Register

Name: CSI2DC_VPROWR
Offset: 0x104
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ROW[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ROW[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ROW[15:0] Row Number
Current active row number of the frame being processed in the video pipe.

42.6.57 CSI2DC Video Pipe Data Type Remap Register

Name: CSI2DC_VPDTRR
Offset: 0x108
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DTRE	ADT[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – DTRE Data Type Remap Enable

Value	Description
0	Data type remap is disabled.
1	Data type remap is enabled.

Bits 5:0 – ADT[5:0] Alternate Data Type
When the remap is activated, the incoming data type is replaced with ADT.

43. Image Sensor Controller (ISC)

43.1 Description

The Image Sensor Controller (ISC) system manages incoming data from a parallel or MIPI sensor. It supports a single active interface. The parallel interface protocol can use a free-running clock or a gated clock strategy. It supports the ITU-R BT 656/1120 422 protocol with a data width of 8 bits or 10 bits and raw Bayer format. The internal image processor includes adjustable white balance, color filter array interpolation, color correction, gamma correction, defective pixel correction, green disparity correction, black level correction, edge adaptive level, horizontal and vertical downscaler, 12 bits to 10 bits compression, programmable color space conversion, horizontal and vertical chrominance subsampling module. The module also integrates a triple channel direct memory access controller host interface.

43.2 Embedded Characteristics

- MIPI CSI2 Interface Supported, Single Component per Clock Cycle or Recommended Memory Storage (RMS) Mode
- Parallel 14-bit Interface for Raw Bayer, YCbCr, Monochrome and JPEG Compressed Sensor Interface
- BT.601/656/1120 Video Interface Supported
- Progressive Systems and Segmented Frame Systems
- Raw Bayer, YCbCr, Luminance (Black and White) Pixel Format Supported
- Resolution up to 3264 x 2464
- Input Pixel Clock up to 266 MHz
- Camera Sensor Clock Generation for Parallel Interface
- Cropping
- Adjustable White Balance
- Raw Bayer Color Filter Array Interpolation
- Color Correction
- Gamma Correction
- Defective Pixel Correction
- Green Disparity Correction
- Black Level Correction
- Edge Adaptive Level
- Vertical And Horizontal Polyphase Downscaler, with 16 Phases and 4 Taps
- Color Space Conversion
- Contrast, Brightness, Hue and Saturation Control
- 4:4:4 to 4:2:2 Subsampler
- 4:2:2 to 4:2:0 Subsampler
- Rounding, Limiting and Packing unit
- Histogram Generation
- System Interface: Direct Memory Access Interface with Packed, Semi Planar and Planar output format
- Output Memory Format: 16 bpp RGB, 32 bpp RGB, 16 bpp, YCbCr 444, YCbCr 422, YCbCr 420, up to 14 bit raw Bayer

43.3 Block Diagram and Use Cases

43.3.1 Functional Diagrams

Figure 43-1. ISC Block Diagram with Active MIPI Interface

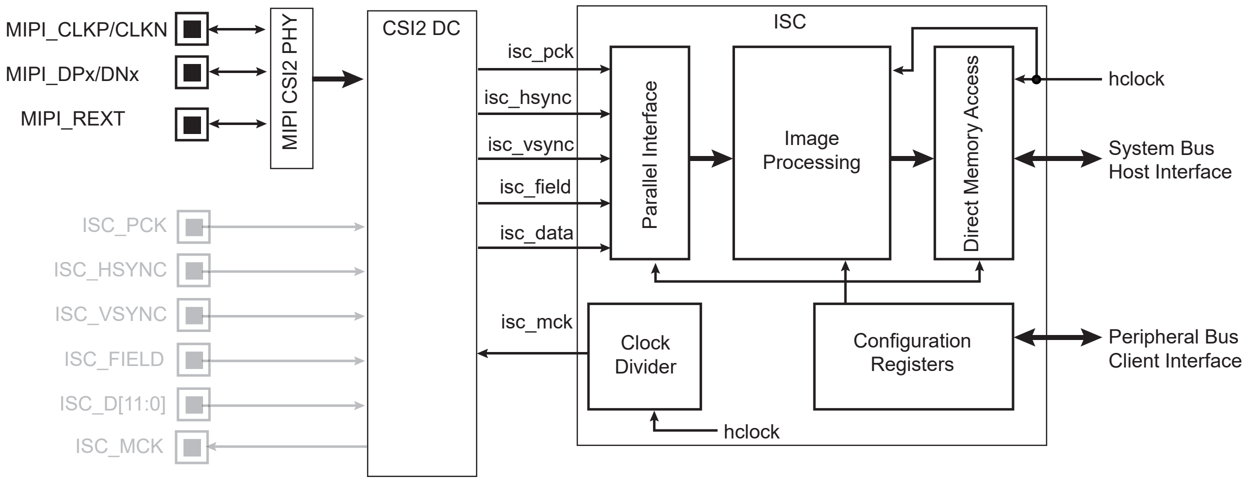


Figure 43-2. ISC Block Diagram with Active Parallel Interface

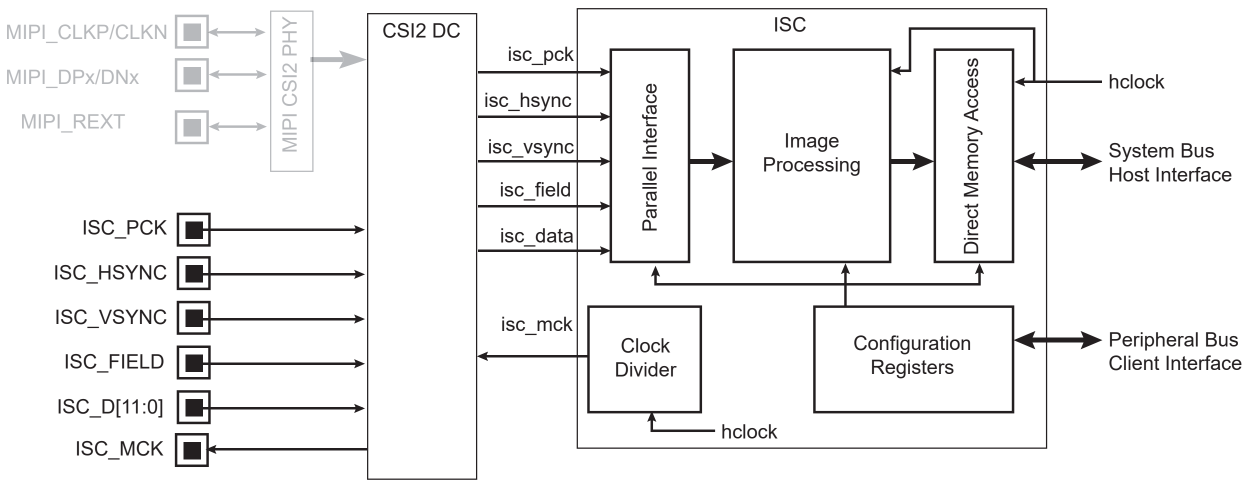
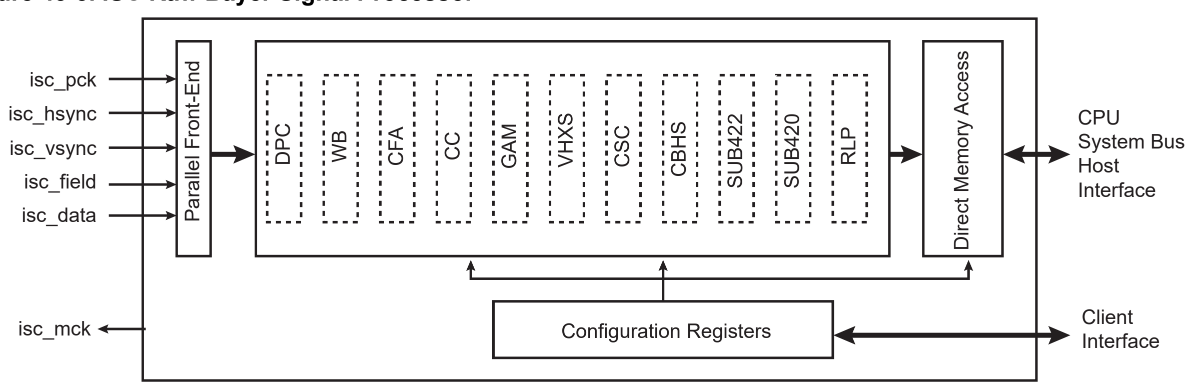


Figure 43-3. ISC Raw Bayer Signal Processor



The ISC video pipeline integrates the following submodules:

- DPC: Defective pixel correction, also including a Green Disparity Correction (GDC), Black Level Correction (BLC), Edge Adaptive Level (EAL)

- PFE: Parallel Front End to sample the camera sensor input stream
- WB: Programmable white balance in the Bayer domain
- CFA: Color filter array interpolation module
- CC: Programmable color correction
- GAM: Gamma correction
- VHXS: Vertical and horizontal scaling engine
- CSC: Programmable color space conversion
- CBHS: Performs contrast, brightness, hue and saturation control
- SUB422: Performs YCbCr444 to YCbCr422 chrominance horizontal subsampling
- SUB420: Performs YCbCr422 to YCbCr420 chrominance vertical subsampling
- RLP: Performs rounding, range limiting and packing of the incoming data.

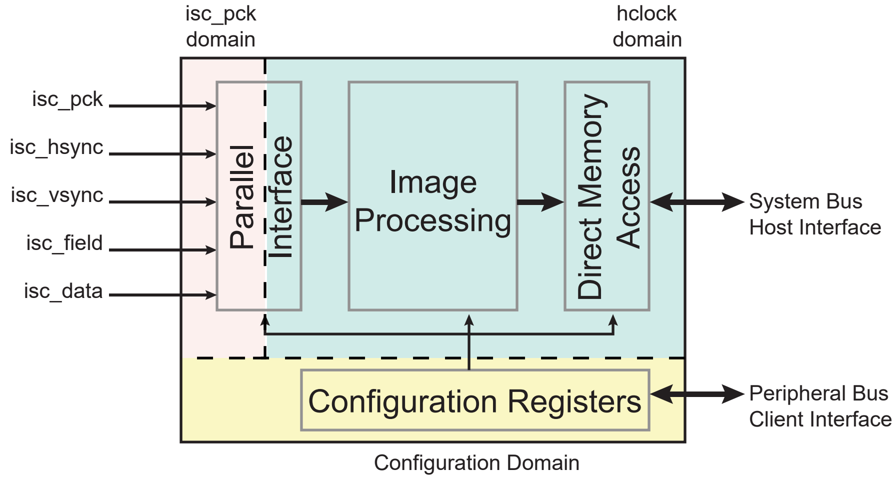
43.4 I/O Lines Description

Table 43-1. I/O Lines Description

Signal Name	Description	Type
ISC_PCK	Image Sensor Pixel clock	Input
ISC_D[11:0]	Image Sensor Data	Input
ISC_VSYNC	Image Sensor Vertical Synchro	Input
ISC_HSYNC	Image Sensor Horizontal Synchro	Input
ISC_FIELD	Field Identification Signal	Input
ISC_MCK	Image Sensor Main clock	Output
MIPI_CLKP	MIPI DPHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI DPHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI DPHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_REXT	Calibration reference resistor	Input/Output

43.4.1 Clock Domain Diagram

Figure 43-4. Clock Domain Hierarchy



43.4.2 Typical Use Cases

Figure 43-5. Raw Bayer Sensor without Embedded Image Processor

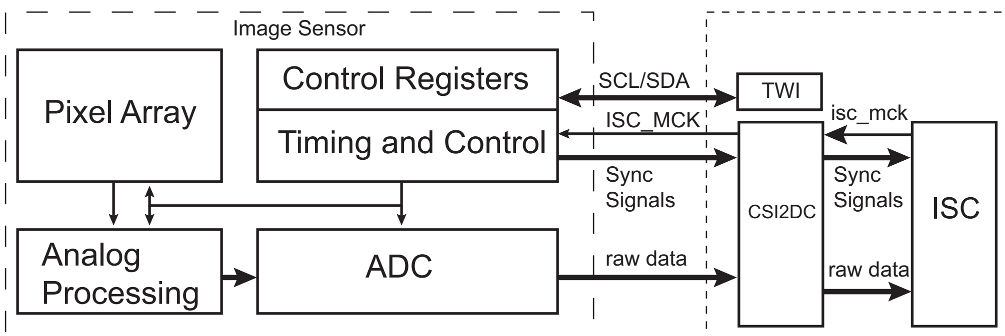


Figure 43-6. MIPI CSI2 Serial Link

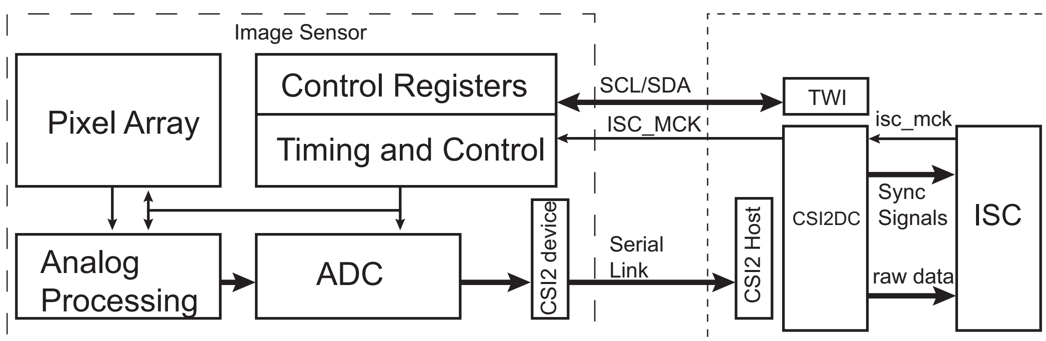


Figure 43-7. Raw Bayer Sensor with Embedded Image Processor

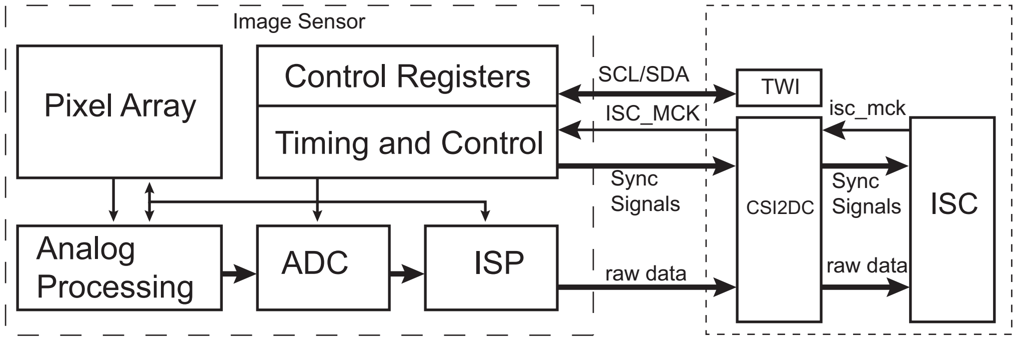


Figure 43-8. BT656 Video Interface Sensor

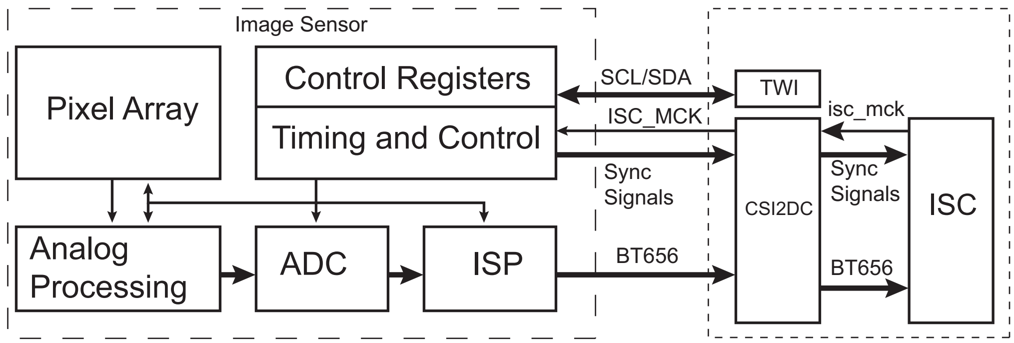


Figure 43-9. Sensor with JPEG Output

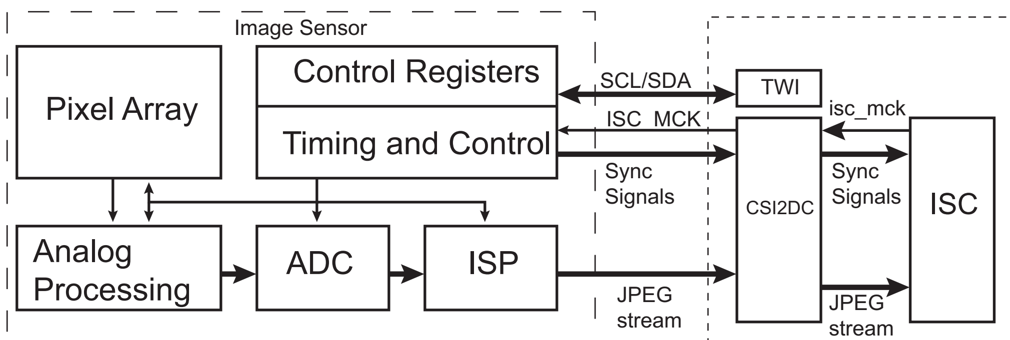
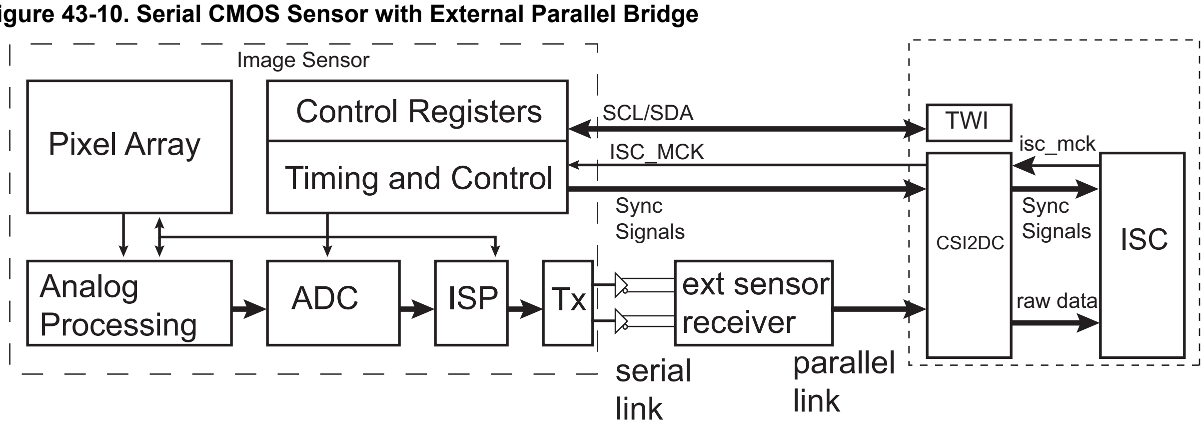


Figure 43-10. Serial CMOS Sensor with External Parallel Bridge



43.5 Product Dependencies

43.5.1 I/O Lines

The parallel interface pins used for interfacing the ISC are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the ISC pins to their peripheral function. If I/O lines of the ISC are not used by the application, they can be used for other purposes by the PIO controller.

When the MIPI link is used to interface the ISC, refer to the section "Camera Serial Interface (CSI)" to enable the MIPI inputs (not multiplexed with PIO lines).

43.5.2 Power Management

The ISC peripheral clock (hclock) is not continuously provided to the ISC. The programmer must first enable the ISC clock in the Power Management Controller (PMC) before using the ISC.

When the MIPI link is used to interface the ISC, refer to the sections "Camera Serial Interface (CSI)" to enable the MIPI PHY and "CSI-2 Demultiplexer Controller (CSI2DC)".

43.5.3 Interrupt Sources

The ISC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the ISC interrupt requires the Interrupt Controller to be programmed first.

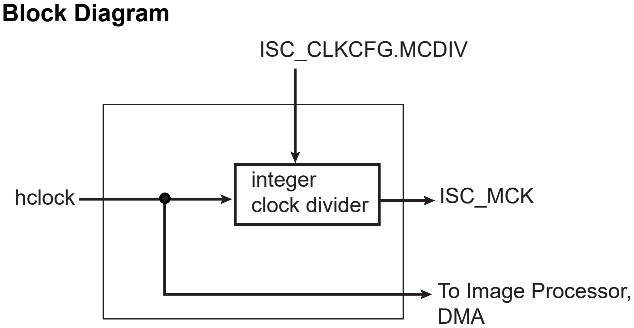
43.6 Functional Description

43.6.1 ISC Clock Management

The ISC module provides the ISC_MCK output clock to the CSI2 Demultiplexer Controller (CSI2DC), which transfers video stream from the MIPI CSI2 clock domain to the ISC_MCK clock domain. See Figure 43-1 and Figure 43-2.

The ISC_MCK clock has one programmable clock divider (ISC_CLKCFG.MCDIV). The clock is enabled using ISC_CLKEN.MCEN.

Figure 43-11. Clock Divider Block Diagram



The ISC is designed to accept input signals that are asynchronous to the hclock.

Synchronization is done internally as long as the following relationship holds:

- isc_pck frequency is lower than or equal to hclock frequency,

43.6.1.1 Software Requirement

A software write operation to the ISC_CLKEN or ISC_CLKDIS register requires double clock domain synchronization and is not permitted when the ISC_CLKSR.SIP is asserted.

43.6.2 Parallel Interface Timing Description

The parallel interface protocol supports two operating modes.

Figure 43-12. Free-Running Pixel Clock

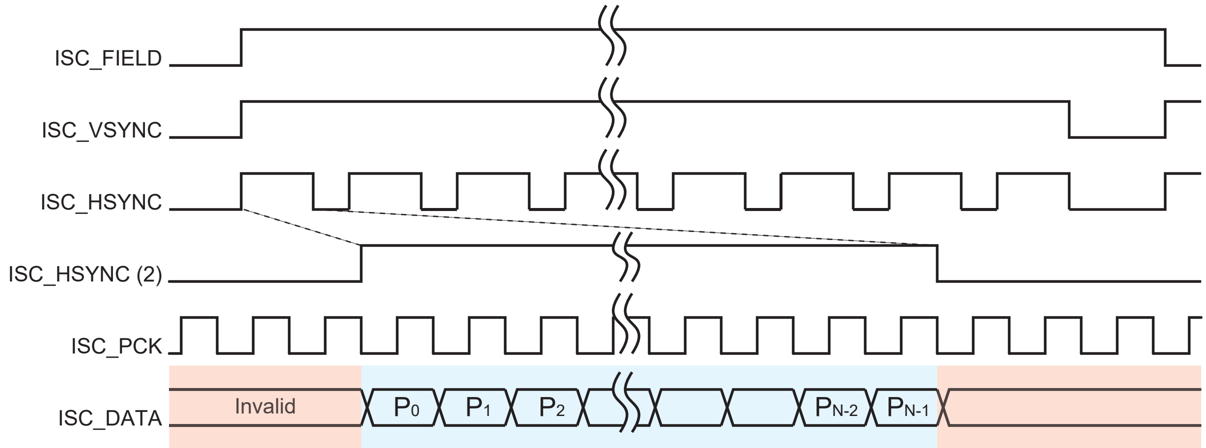
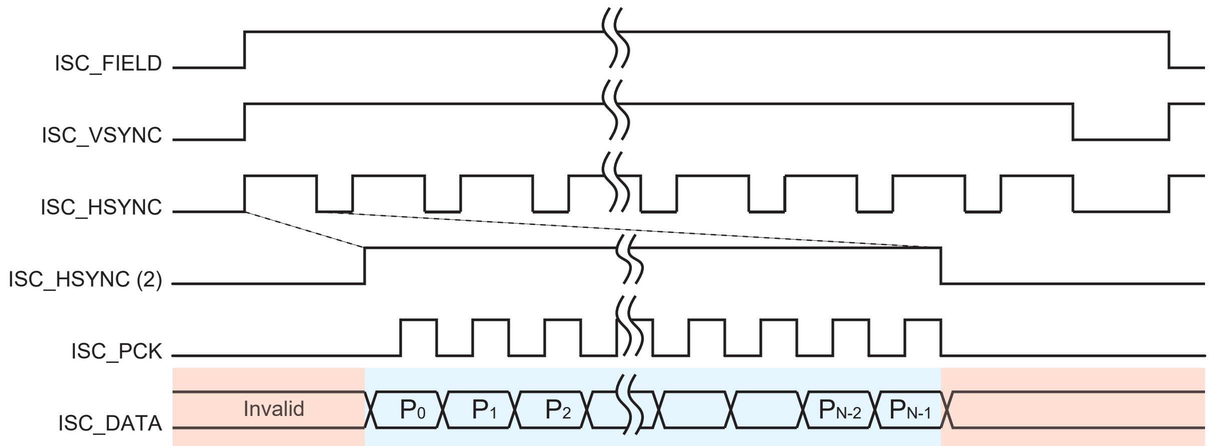


Figure 43-13. Gated Pixel Clock



43.6.3 BT.601/656/1120 Embedded Timing Synchronization Operation

The ISC module supports embedded synchronization decoding. When the ISC_PFE_CFG0.CCIR656 is set, the decoder is activated and signals isc_vsync and isc_hsync are not used to decode the valid pixels. If the CCIR10_8N is set, the bitstream is 10 bits wide, otherwise it is only 8 bits wide. When the ISC_PFE_CFG0.CCIR_CRC is set, the decoder automatically corrects the error.

Figure 43-14. Field/Segment Timing Relationship for Interlaced and Segmented Frame Systems

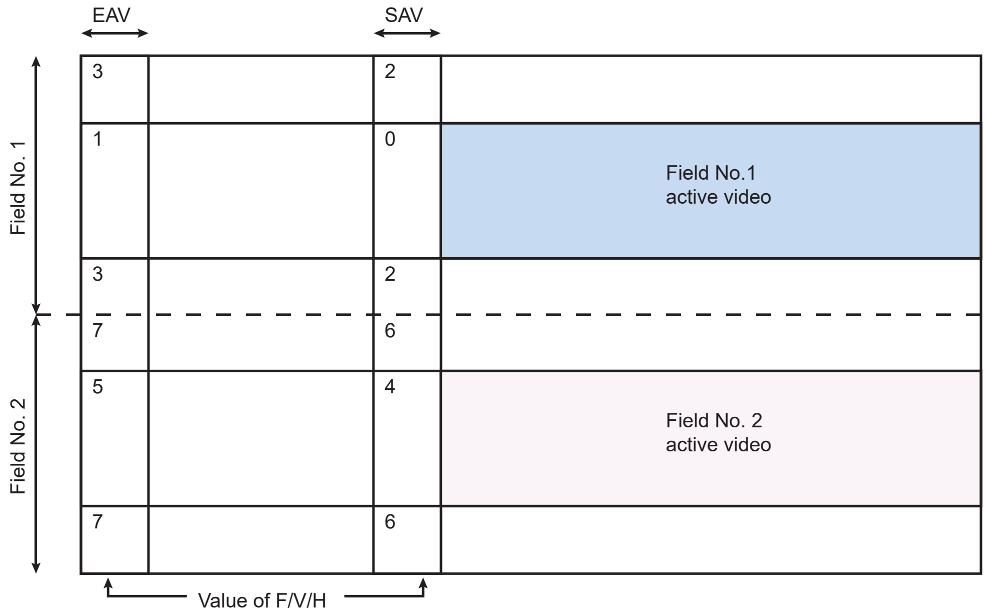
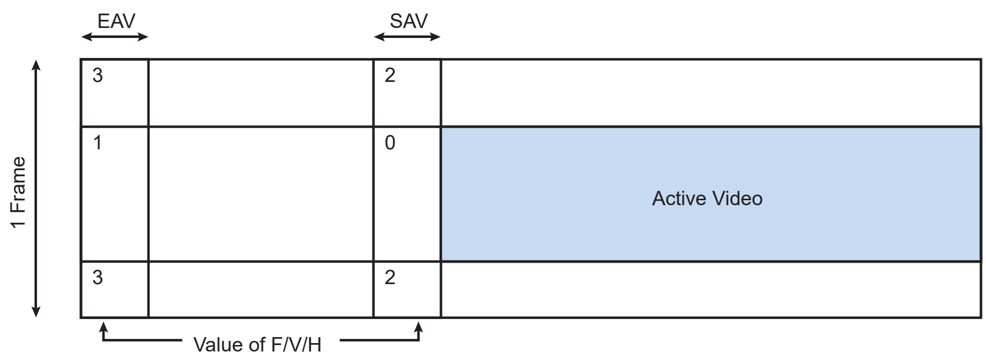


Figure 43-15. Frame Timing Relationship for Progressive Systems



43.6.4 Parallel Interface External Sensor Connections

43.6.4.1 YCbCr, 10-bit CCIR656 with Embedded Synchronization

This mode is activated when ISC_PFE_CFG0.CCIR656 and ISC_PFE_CFG0.CCIR10_8N are both set.

Interface Bit	First Word	Second Word	Third Word	Fourth Word
ISC_DATA[11](MSB)	1	0	0	1
ISC_DATA[10]	1	0	0	F
ISC_DATA[9]	1	0	0	V
ISC_DATA[8]	1	0	0	H
ISC_DATA[7]	1	0	0	P3
ISC_DATA[6]	1	0	0	P2
ISC_DATA[5]	1	0	0	P1
ISC_DATA[4]	1	0	0	P0
ISC_DATA[3]	1	0	0	0

...........continued				
Interface Bit	**First Word**	**Second Word**	**Third Word**	**Fourth Word**
ISC_DATA[2]	1	0	0	0
ISC_DATA[1]	Not Used	Not Used	Not Used	Not Used
ISC_DATA[0]	Not Used	Not Used	Not Used	Not Used

43.6.4.2 YCbCr, 8-bit CCIR656 with Embedded Synchronization

This mode is activated when ISC_PFE_CFG0.CCIR656 is set and ISC_PFE_CFG0.CCIR10_8N is cleared.

Interface Bit	First Word	Second Word	Third Word	Fourth Word
ISC_DATA[11](MSB)	1	0	0	1
ISC_DATA[10]	1	0	0	F
ISC_DATA[9]	1	0	0	V
ISC_DATA[8]	1	0	0	H
ISC_DATA[7]	1	0	0	P3
ISC_DATA[6]	1	0	0	P2
ISC_DATA[5]	1	0	0	P1
ISC_DATA[4]	1	0	0	P0
ISC_DATA[3]	Not Used	Not Used	Not Used	Not Used
ISC_DATA[2]	Not Used	Not Used	Not Used	Not Used
ISC_DATA[1]	Not Used	Not Used	Not Used	Not Used
ISC_DATA[0]	Not Used	Not Used	Not Used	Not Used

43.6.4.3 Raw Bayer Parallel Interface

The table below shows how to connect the data bus of a raw Bayer sensor.

Interface	Bayer 12-bit	Bayer 11-bit	Bayer 10-bit	Bayer 9-bit	Bayer 8-bit
ISC_DATA[11](MSB)	DOUT[11]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]
ISC_DATA[10]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]
ISC_DATA[9]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]
ISC_DATA[8]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]	DOUT[4]
ISC_DATA[7]	DOUT[7]	DOUT[6]	DOUT[5]	DOUT[4]	DOUT[3]
ISC_DATA[6]	DOUT[6]	DOUT[5]	DOUT[4]	DOUT[3]	DOUT[2]
ISC_DATA[5]	DOUT[5]	DOUT[4]	DOUT[3]	DOUT[2]	DOUT[1]
ISC_DATA[4]	DOUT[4]	DOUT[3]	DOUT[2]	DOUT[1]	DOUT[0]
ISC_DATA[3]	DOUT[3]	DOUT[2]	DOUT[1]	DOUT[0]	Not Used
ISC_DATA[2]	DOUT[2]	DOUT[1]	DOUT[0]	Not Used	Not Used
ISC_DATA[1]	DOUT[1]	DOUT[0]	Not Used	Not Used	Not Used
ISC_DATA[0]	DOUT[0]	Not Used	Not Used	Not Used	Not Used

43.6.4.4 Monochrome Parallel Interface

The table below shows how to connect the data bus of a Monochrome sensor.

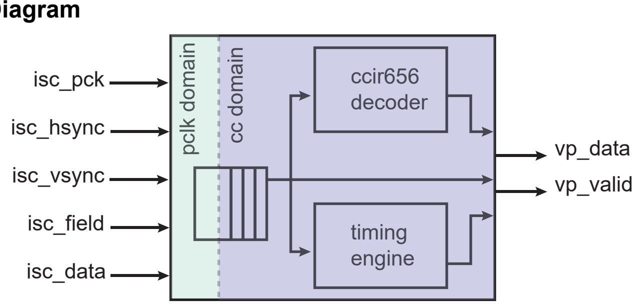
Interface	Mono 12-bit	Mono 11-bit	Mono 10-bit	Mono 9-bit	Mono 8-bit
ISC_DATA[11](MSB)	DOUT[11]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]
ISC_DATA[10]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]
ISC_DATA[9]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]
ISC_DATA[8]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]	DOUT[4]
ISC_DATA[7]	DOUT[7]	DOUT[6]	DOUT[5]	DOUT[4]	DOUT[3]
ISC_DATA[6]	DOUT[6]	DOUT[5]	DOUT[4]	DOUT[3]	DOUT[2]
ISC_DATA[5]	DOUT[5]	DOUT[4]	DOUT[3]	DOUT[2]	DOUT[1]
ISC_DATA[4]	DOUT[4]	DOUT[3]	DOUT[2]	DOUT[1]	DOUT[0]
ISC_DATA[3]	DOUT[3]	DOUT[2]	DOUT[1]	DOUT[0]	Not Used
ISC_DATA[2]	DOUT[2]	DOUT[1]	DOUT[0]	Not Used	Not Used
ISC_DATA[1]	DOUT[1]	DOUT[0]	Not Used	Not Used	Not Used
ISC_DATA[0]	DOUT[0]	Not Used	Not Used	Not Used	Not Used

43.6.5 MIPI Interface Mapping

When operating with a MIPI interface, CSI2DC maps the pixel content on the input data bus according to the data format. Refer to Functional Description in the section “CSI-2 Demultiplexer Controller (CSI2DC)”.

43.6.6 Parallel Front End (PFE) Module

Figure 43-16. PFE Block Diagram



The Parallel Front End module performs data resampling across clock domain boundary. It includes a CCIR656 decoder used to convert a standard ITU-R BT.656 stream to 24-bit digital video. It also generates pixels, syncs flags and valid signals to the main video pipeline. It ouputs field, video and synchronization signals. The PFE can optionally crop and limit the incoming pixel stream to a predefined horizontal and vertical value. By default, the PFE only relies on the input horizontal and vertical references to sample the incoming pixel stream. A pixel is sampled if, and only if, the vertical and horizontal synchronizations are valid and a pixel clock edge is detected. ISC_PFE_CFG0.BPS shows the number of bits per sample.

When operating with a parallel interface, the PFE module outputs a 12-bit data on the vp_data[11:0] bus, and asserts the vp_valid signal when the data can be sampled

PFE vp_data Mapping	Raw Bayer 12-bit	Raw Bayer 10-bit	YUV422 8-bit	YUV422 10-bit	Mono 12-bit
vp_data[39:12]	–	–	–	–	–
vp_data[11]	RGGB[11]	RGGB[9]	YC422[7]	YC422[9]	Y[11]

...........continued					
PFE vp_data Mapping	**Raw Bayer** 12-bit	**Raw Bayer** 10-bit	**YUV422** 8-bit	**YUV422** 10-bit	**Mono** 12-bit
vp_data[10]	RGGB[10]	RGGB[8]	YC422[6]	YC422[8]	Y[10]
vp_data[9]	RGGB[9]	RGGB[7]	YC422[5]	YC422[7]	Y[9]
vp_data[8]	RGGB[8]	RGGB[6]	YC422[4]	YC422[6]	Y[8]
vp_data[7]	RGGB[7]	RGGB[5]	YC422[3]	YC422[5]	Y[7]
vp_data[6]	RGGB[6]	RGGB[4]	YC422[2]	YC422[4]	Y[6]
vp_data[5]	RGGB[5]	RGGB[3]	YC422[1]	YC422[3]	Y[5]
vp_data[4]	RGGB[4]	RGGB[2]	YC422[0]	YC422[2]	Y[4]
vp_data[3]	RGGB[3]	RGGB[1]	YC422[7] or 0	YC422[1]	Y[3]
vp_data[2]	RGGB[2]	RGGB[0]	YC422[6] or 0	YC422[0]	Y[2]
vp_data[1]	RGGB[1]	RGGB[9] or 0	YC422[5] or 0	YC422[9] or 0	Y[1]
vp_data[0]	RGGB[0]	RGGB[8] or 0	YC422[4] or 0	YC422[8] or 0	Y[0]

Note: When ISC_PFE_CFG0.REP is set, missing vp_data LSBs are replaced with replicated LSBs of the incoming stream, otherwise they are forced to zero.

When operating with MIPI RGB or YUV formats, the PFE module copies up to 40-bit data from isc_data[39:0] bus on the vp_data[39:0] bus, and asserts the vp_valid signal when the data can be sampled.

The PFE module also includes logic to synchronize capture request with the incoming pixel stream. Two operating modes are available: Single Shot and Continuous Acquisition. When the ISC_PFE_CFG0.CONT is cleared, the ISC transfers a single image to memory,

Figure 43-17. Single Shot Mode

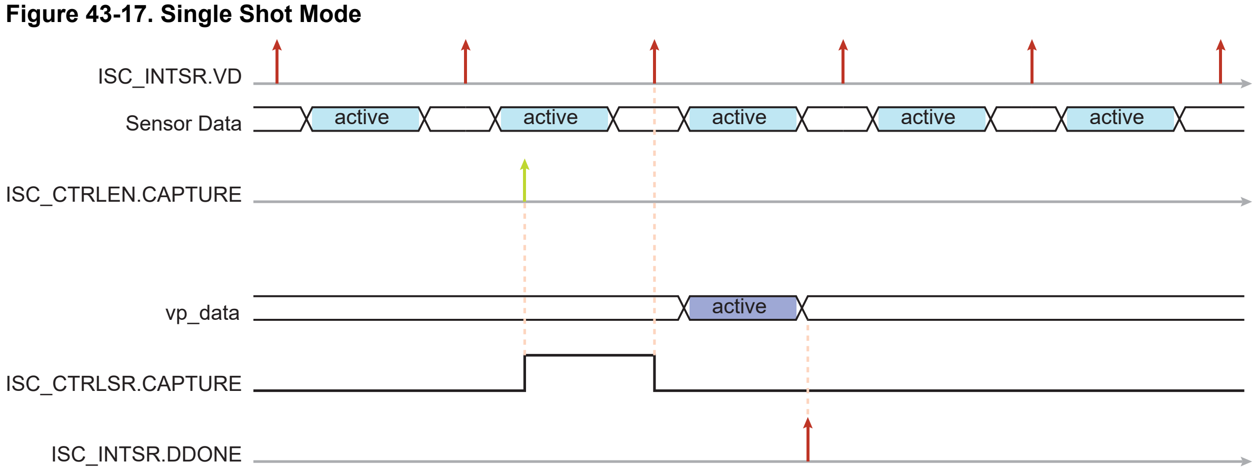
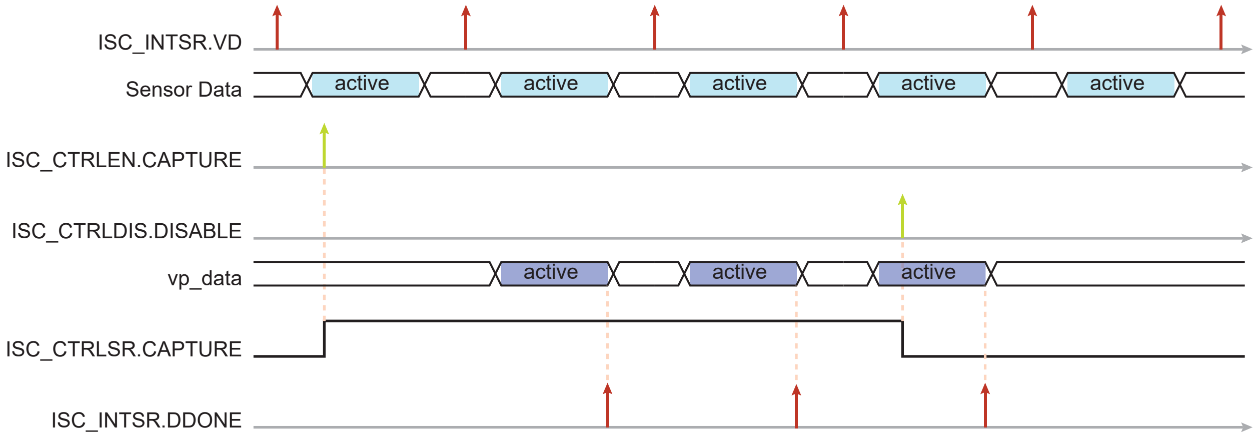
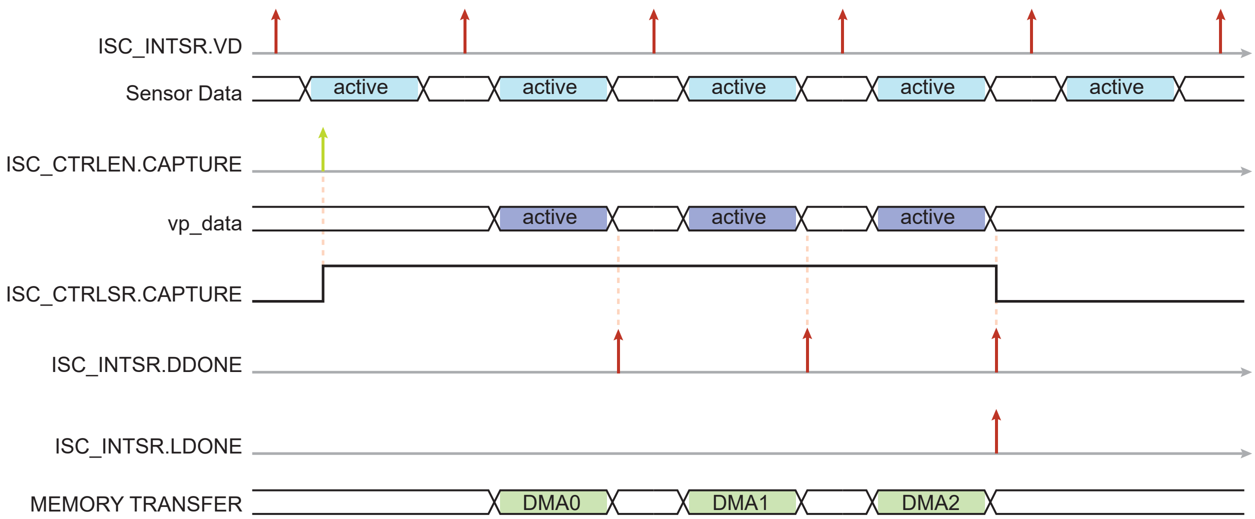


Figure 43-18. Continuous Acquisition Mode



When Continuous Acquisition mode is activated (ISC_PFE_CFG0.CONT is set), the data transfer terminates when either a DMA end of list is reached, a software disable is performed or a software reset is activated. The ISC_INTSR.DDONE is set at the end of the DMA data transfer.

Figure 43-19. Continuous Acquisition, DMA Terminated



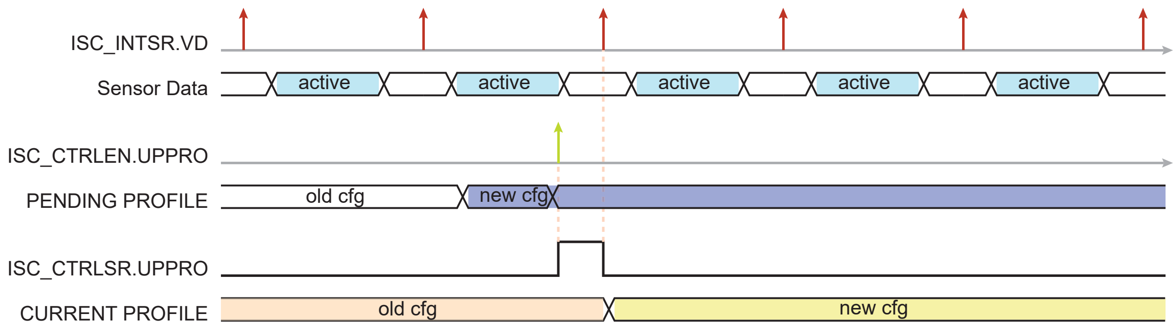
The linked list DMA transfer is terminated when an item of the list is programmed with ISC_DCTRL.DE cleared or when ISC_DNDA.NDA is equal to zero. This configuration also clears ISC_CTRLSR.CAPTURE and sets the ISC_INTSR.LDONE interrupt flag.

The linked list DMA transfer starts if ISC_DCTRL.DE is set and if ISC_DNDA.NDA is different from zero.

43.6.6.1 Update the ISC Profile

Each ISC register is double-buffered to simplify the software configuration and the synchronization with the associated frame buffer. When the configuration of the ISC is modified, ISC_CTRLEN.UPPRO must be set to transfer the configuration from the input buffer to the ISC video pipeline.

Figure 43-20. Update Profile Timing Diagram



43.6.6.2 Software Requirements

Writing to the ISC_CTRLEN or ISC_CTRLDIS register requires a double domain synchronization, so it is forbidden to write these registers when ISC_CTRLSR.SIP is asserted.

43.6.7 Defective Pixel Correction (DPC)

The ISC is able to detect and correct defective pixels.

The defective pixel correction is enabled by writing '1' to ISC_DPC_CTRL.DPCEN. The color filter array pattern of the image sensor driven by ISC must be configured in ISC_DPC_CFG.BAYCFG.

It is possible to adjust the processing performed by the defective pixel correction logic.

Several criteria can be activated to determine if a pixel is defective (edge interpolation, median threshold, closest pixel, average threshold).

The edge interpolation is enabled by writing ISC_DPC_CFG.EITPOL=1.

The median threshold is enabled by writing ISC_DPC_CFG.TM_ENABLE=1.

The closest pixel threshold is enabled by writing ISC_DPC_CFG.TC_ENABLE=1.

The average threshold is enabled by writing ISC_DPC_CFG.TA_ENABLE=1.

The different thresholds can be configured in ISC_DPC_THRESHM.THRESHM, ISC_DPC_THRESHC.THRESHC, ISC_DPC_THRESHA.THRESHA.

The triggering of a pixel correction can be triggered when all active criteria are met or only one is met by configuring the bit ISC_DPC_CFG.ND_MODE.

The number of corrected pixels by frame are reported in ISC_DPC_SR.COUNTER.

43.6.8 Green Disparity Correction (GDC)

The color filter array pattern alternates horizontal lines made of green and blue pixels with lines made of green and red pixels.

This pattern introduces green disparity from line to line that can automatically corrected by writing a '1' in ISC_DPC_CTRL.GDCEN.

The green disparity clipping value must be configured in ISC_DPC_CFG.GDCCLP.

43.6.9 Black Level Correction (BLC)

The black level provided by the image sensor is automatically corrected if ISC_DPC_CTRL.BLCEN=1.

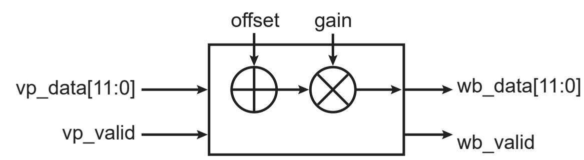
The black level offset provided by the image sensor (refer to the image sensor reference manual) must be configured in ISC_DPC_CFG.BLOFST.

43.6.10 White Balance (WB) Module

The White Balance (WB) module captures the vp_data bus from the PFE module when the vp_valid signal is asserted, and it generates a wb_data data along with its validity signal wb_valid.

When operating with Raw Bayer formats, and ISC_WB_CTRL.ENABLE is set, each Bayer color component (R, Gr, B, Gb) can be manually adjusted using an offset and a gain. The Bayer pattern is adjustable using ISC_WB_CFG.BAYCFG.

Figure 43-21. WB Block Diagram



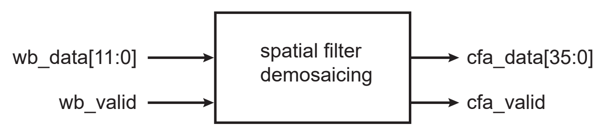
There are four {gain, offset} sets for each component. The output value is clipped.

ISC_WB_CTRL.ENABLE	WB_DATA Slice	Value
0	wb_data[11:0]	vp_data[11:0]
1	wb_data[11:0]	clipped((vp_data[11:0]+offset)*gain)

43.6.11 Color Filter Array (CFA) Interpolation Module

In a single-sensor system, each cell on the sensor has a specific color filter and microlens positioned above it. The raw data obtained from the sensor do not have the full R/G/B information at each cell position. Color interpolation is required to retrieve the missing components. The CFA module samples the wb_data[11:0] 12-bit bus when wb_valid is asserted and generates a 36-bit width data bus cfa_data[35:0] with the validity bit cfa_valid.

Figure 43-22. CFA Block Diagram



ISC_CFA_CTRL.ENABLE	CFA_DATA Slice	Value
0	cfa_data[35:24]	wb_data[11:0]
	cfa_data[23:12]	wb_data[11:0]
	cfa_data[11:0]	wb_data[11:0]
1	cfa_data[35:24]	R = spatial_filter_R(wb_data[11:0])
	cfa_data[23:12]	G = spatial_filter_G(wb_data[11:0])
	cfa_data[11:0]	B = spatial_filter_B(wb_data[11:0])

The filter kernel size is 5, and requires two additional lines to initialize the filter. When ISC_CFA_CFG.EITPOL is set, the missing information is interpolated from the nearest neighbor. If ISC_CFA_CFG.EITPOL is cleared, only valid pixels are used to initialize the filter kernel, but the output number of lines is less than the input number of lines. In that case, four lines are consumed to fill the kernel.

43.6.11.1 Frame Size Requirement when Edge Interpolation is Off, ISC_CFA_CFG.EITPOL Cleared

- Minimum number of rows (in): 5
- Minimum number of columns (in): 5
- Number or rows after CFA: Number of rows (in) - 4
- Number of columns after CFA: Number of columns (in) - 4

43.6.11.2 Frame Size Requirement when Edge Interpolation is On, ISC_CFA_CFG.EITPOL Set

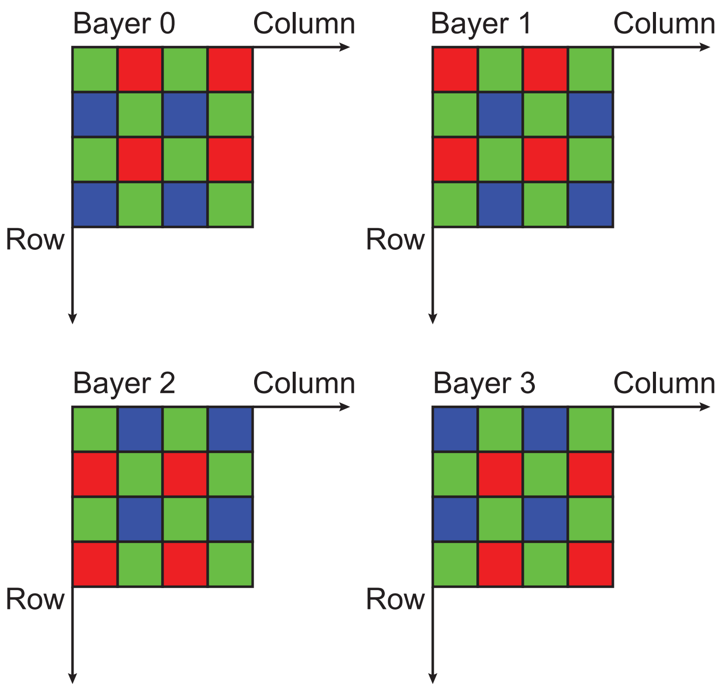
- Minimum number of rows (in): 3
- Minimum number of columns (in): 3
- Number of rows after CFA: Number of rows (in)
- Number of columns after CFA: Number of columns (in)

43.6.11.3 Bayer Mode and Edge Interpolation Description

When Edge Interpolation mode (ISC_CFA_CFG.EITPOL) is activated, dummy lines are generated using rows and columns replication.

The CFA module supports four sensor alignments using ISC_CFA_CFG.BAYCFG. See the figure below.

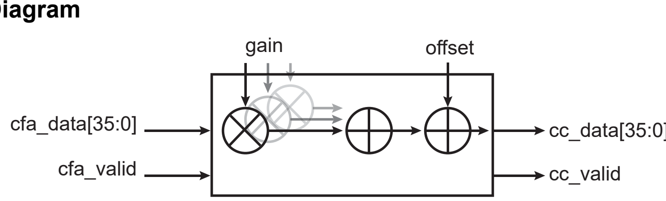
Figure 43-23. Supported Color Filter Array Patterns



43.6.12 Color Correction (CC) Module

RGB color correction is used to compensate for cross color bleeding in the filter used with the image sensor. The module samples the cfa_data[35:0] 36-bit bus when cfa_valid is asserted and generate a cc_data[35:0] 36-bit wide bus and a cc_valid signal.

Figure 43-24. CC Block Diagram



There are three {gain, offset} sets for color component R, G, B.

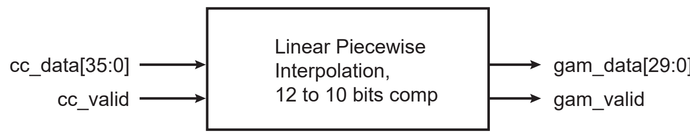
ISC_CC_CTRL.ENABLE	CC_DATA Slice	Value
0	cc_data[35:24]	cfa_data[11:0]
	cc_data[23:12]	cfa_data[11:0]
	cc_data[11:0]	cfa_data[11:0]
1	cc_data[35:24]	R=clipped(sum(cfa_data_x * gain_Rx) + offset_R)
	cc_data[23:12]	G=clipped(sum(cfa_data_x * gain_gx) + offset_g)
	cc_data[11:0]	B=clipped(sum(cfa_data_x * gain_Bx) + offset_B)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \text{RRGAIN} & \text{RGGAIN} & \text{RBGAIN} \\ \text{GRGAIN} & \text{GGGAIN} & \text{GBGAIN} \\ \text{BRGAIN} & \text{BGGAIN} & \text{BBGAIN} \end{bmatrix} \times \begin{bmatrix} \text{cfa\_data}[35:24] \\ \text{cfa\_data}[23:12] \\ \text{cfa\_data}[11:0] \end{bmatrix} + \begin{bmatrix} \text{ROFST} \\ \text{GOFST} \\ \text{BOFST} \end{bmatrix}$$

43.6.13 Gamma Curve (GAM) Module

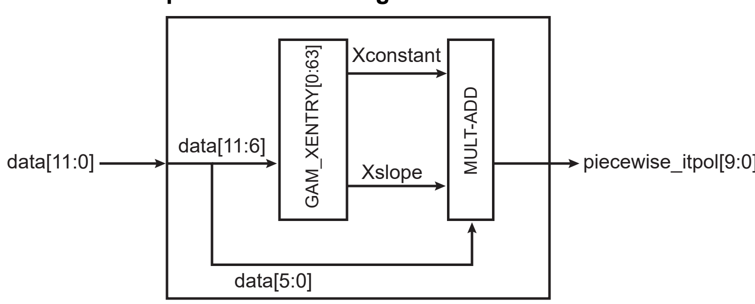
The GAM module samples the cc_data[35:0] bus when cc_valid is asserted, and generates gam_data[29:0] 30-bit width data along with the validity signal gam_valid. Imaging devices have non-linear characteristics, but the transfer function is approximated by a power function. The intensity of each of the linear RGB components is transformed to a non-linear signal through the use of the gamma correction submodule. The power function is linearly interpolated using 64 breakpoints. This also performs a 12-bit to 10-bit compression. The polynomial for the linear interpolation between breakpoints is i and i +1. Consequently, for each breakpoint, two values are required: constant and slope. The table values are programmable through the user interface when the gamma correction module is disabled (ISC_GAM_CTRL.ENABLE is cleared). ISC_GAM_RENTRY is used for Red gamma correction. ISC_GAM_GENTRY is used for Green gamma correction. ISC_GAM_BENTRY is used for Blue gamma correction. Each table entry is composed of a 10-bit (signed) slope and a 10-bit constant.

Figure 43-25. GAM Block Diagram



ISC_GAM_CTRL.ENABLE	ISC_GAM_CTRL.XLUT	GAM_DATA Slice	Value
0	0	gam_data[29:0]	cc_data[29:0]
1	0	gam_data[29:20]	cc_data[35:26]
		gam_data[19:10]	cc_data[23:14]
		gam_data[9:0]	cc_data[11:2]
1	1	gam_data[29:20]	R=piecewise_itpol(cc_data_r[35:24])
		gam_data[19:10]	G=piecewise_itpol(cc_data_r[23:12])
		gam_data[9:0]	B=piecewise_itpol(cc_data_r[11:0])

Figure 43-26. Piecewise Linear Interpolation Block Diagram

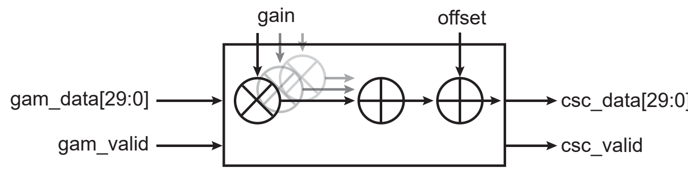


The interpolation consists of three tables that store the function values GAM_XENTRY[0:63] where X stands for R, G and B. The input of the table has six bits. It outputs a slope and a constant. The slope is later multiplied by the data lsb (6-bit) and added to a constant. The final value is the gamma-corrected value of the input. This module performs a 12-to-10 compression.

43.6.14 Color Space Conversion (CSC) Module

By converting an image from RGB to YCbCr color space, it is possible to separate Y, Cb and Cr information. The CSC samples the gam_data[29:0] 30-bit data bus, extracts YCbCr information from the sampled data, and then generates the color-converted data csc_data[29:0] and the validity signal csc_valid.

Figure 43-27. CSC Block Diagram



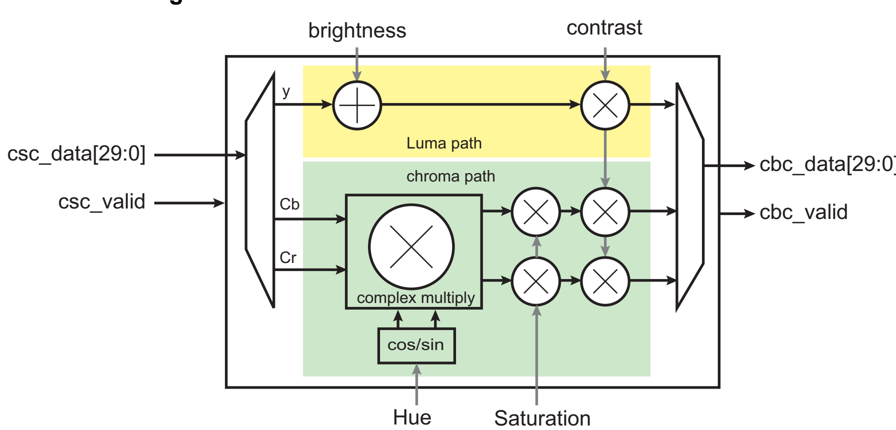
ISC_CSC_CTRL.ENABLE	CSC_DATA Slice	Value
0	csc_data[29:0]	gam_data[29:0]
1	csc_data[29:20]	Y = clipped(sum(gam_data_x * gain_Yx) + offset_y << 2)
	csc_data[19:10]	Cb = clipped(sum(gam_data_x * gain_Cbx)+offset_cb << 2)
	csc_data[9:0]	Cr = clipped(sum(gam_data_x * gain_Crx)+offset_cr << 2)

$$\begin{bmatrix} Y \\ \mathrm{CB} \\ \mathrm{CR} \end{bmatrix} = \begin{bmatrix} \mathrm{YR} & \mathrm{YG} & \mathrm{YB} \\ \mathrm{CBR} & \mathrm{CBG} & \mathrm{CBB} \\ \mathrm{CRR} & \mathrm{CRG} & \mathrm{CRB} \end{bmatrix} \times \begin{bmatrix} \mathrm{gam\_data}[29:20] \\ \mathrm{gam\_data}[19:10] \\ \mathrm{gam\_data}[9:0] \end{bmatrix} + \begin{bmatrix} \mathrm{YOFST} \\ \mathrm{CBOFST} \\ \mathrm{CROFST} \end{bmatrix}$$

43.6.15 Contrast, Brightness, Hue and Saturation

This module is for YUV formats purpose. Brightness Offset allows the Luminance to be adjusted. Hue is used for Chroma phase adjustment, and Color Saturation for Chroma amplitude. Contrast gain is applied on all pixel components (Luma and Chroma). The CBHS samples the csc_data[29:0] 30-bit bus when csc_valid is asserted and generates cbhs_data[29:0] with the validity signal cbc_valid.

Figure 43-28. CBHS Block Diagram



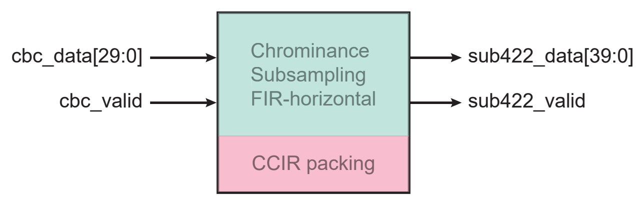
ISC_CBC_CTRL.ENABLE	ISC_CBC_CFG.CCIR	CBC_DATA Slice	Value
0	0	cbc_data[29:0]	csc_data[29:0]

...........continued			
ISC_CBC_CTRL.ENABLE	ISC_CBC_CFG.CCIR	CBC_DATA Slice	Value
1	0	cbc_data[29:20]	Y = clipped(contrast * (csc_data[29:20]+brightness))
		cbc_data[19:10]	Cb = clipped(saturation * contrast * (csc_data[19:10]*cos(hue) +csc_data[9:0]*sin(hue))
		cbc_data[9:0]	Cr = clipped(saturation * contrast * (csc_data[19:10]*sin(hue)-csc_data[9:0]*cos(hue))
1	1	cbc_data[29:10]	0
		cbc_data[9:0]	ccir656 stream with luminance correction

43.6.16 4:4:4 To 4:2:2 Chrominance Horizontal Subsampler (SUB422) Module

The color space conversion output stream is a full-bandwidth YCbCr 4:4:4 signal. The chrominance subsampling divides the horizontal chrominance sampling rate by two. A horizontal low pass filter is applied to avoid aliasing effect. The SUB422 module samples 444 full scale YCbCr cbc_data[29:0] 30-bit data, performs horizontal subsampling and generates the sub422_data[39:0] 40-bit data bus with its validity signal sub422_valid.

Figure 43-29. SUB422 Block Diagram



ISC_SUB422_CTRL.ENABLE	ISC_SUB422_CFG.CCIR	SUB422_DATA Slice	Value
0	0	sub422_data[29:0]	cbc_data[29:0]
1	0	sub422_data[39:30]	Y1= cbc_data1[29:20]
		sub422_data[29:20]	Y0 = cbc_data0[29:20]
		sub422_data[19:10]	Cb = filter_hor(cbc_data[19:10])
		sub422_data[9:0]	Cr = filter_hor(cbc_data[9:0])
1	1	sub422_data[39:30]	Y1 = cbc_data[9:0]
		sub422_data[29:20]	Y0 = cbc_data[9:0]
		sub422_data[19:10]	Cb = cbc_data[9:0]
		sub422_data[9:0]	Cr = cbc_data[9:0]

The filter_hor function included in the sub422 module is the chrominance horizontal filter.

sub422 Data Slice	YCbCr Mapping
sub422_data[39:30]	Y1 (sample n)
sub422_data[29:20]	Y0 (sample n-1)
sub422_data[19:10]	Cb (from filter)
sub422_data[9:0]	Cr (from filter)

The filter chrominance position is selectable through the use of ISC_SUB422_CFG.FILTER.

Figure 43-30. Cosited Filter Configuration

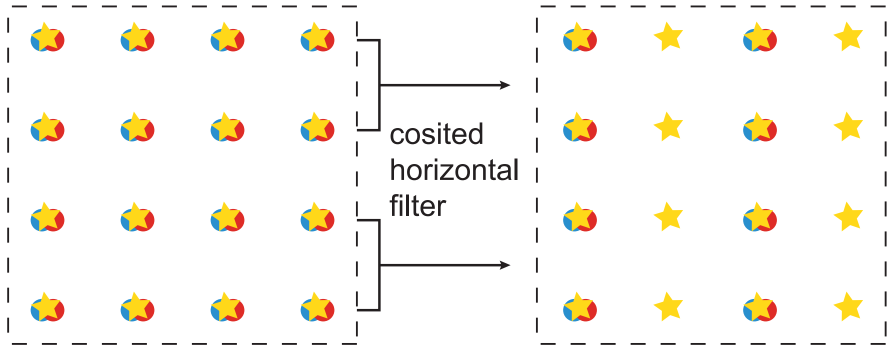
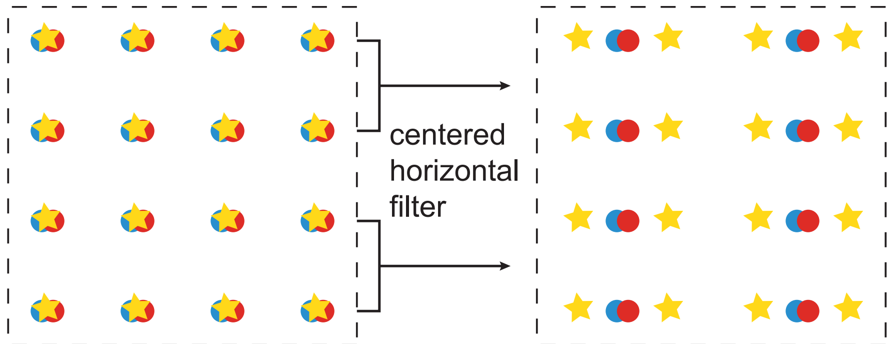


Figure 43-31. Centered Filter Configuration



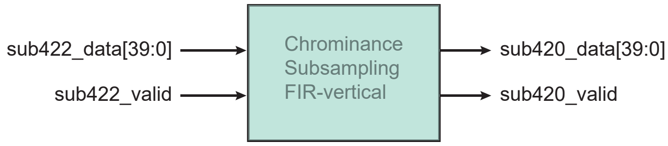
The SUB422 module performs luminance and chrominance packing. When the line length is odd, the missing luminance is a copy of the last but one luminance. It also means that the final dma stream written to memory is equal to the original horizontal size plus one when the line length is odd.

SUB422_DATA Slice	Even Line Length	Odd Line Length
sub422_data[39:30]	Y(n)	Y(n-1)
sub422_data[29:20]	Y(n-1)	Y(n-1)
sub422_data[19:10]	Cb (filtered)	Cb (filtered)
sub422_data[9:0]	Cr (filtered)	Cr (filtered)

43.6.17 4:2:2 To 4:2:0 Chrominance Vertical Subsampler (SUB420) Module

The chrominance subsampling divides the vertical chrominance sampling rate by two. A vertical low pass filter is applied to avoid aliasing effect. Two different filters are used when the source frame is interlaced, and the filter configuration depends on the field value (the field is propagated in the video pipeline).

Figure 43-32. SUB420 Block Diagram



The SUB420 module samples the sub422_data[39:0] 40-bit data when sub422_valid is asserted, then it performs a vertical subsampling and generates a valid sub420_data[39:0] 40-bit word and the corresponding sub420_valid signal.

ISC_CFA_CTRL.ENABLE	SUB420_DATA Slice	Value
0	sub420_data[39:0]	sub422_data[39:0]

...........continued

ISC_CFA_CTRL.ENABLE	SUB420_DATA Slice	Value
1	sub420_data[39:30]	Y1 = sub422_data[39:30]
	sub420_data[29:20]	Y0 = sub422_data[29:20]
	sub420_data[19:10]	Cb = filter_ver(sub422[19:10])
	sub420_data[9:0]	Cr = filter_ver(sub422[9:0])

The vertical filter is a two-tap filter; for progressive content the coefficient i {1, 1}. When an interlaced field is downsampled, the coefficients are different between the top and the bottom fields.

Figure 43-33. Vertical Chrominance Filter for Progressive Content (Cosited Chrominance Example)

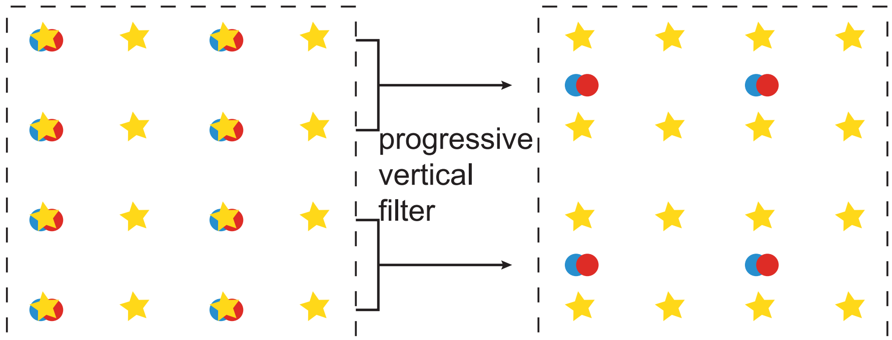


Figure 43-34. Field-dependent Chrominance Filter for Interlaced Content (Cosited Chrominance Example)

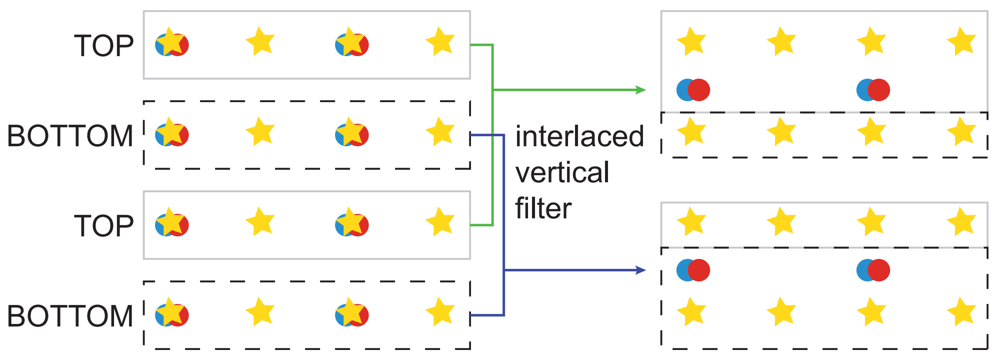


Table 43-2. Filter Configuration

ISC_SUB420_CTRL.FILTER	Field	Filter Configuration
0	progressive	{1, 1}
1	0 (TOP)	{3, 1}
	1 (BOTTOM)	{1, 3}

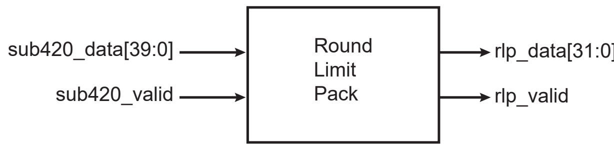
Table 43-3. Output Line Length Configuration

SUB420 Input Number of Rows	SUB420 Luminance Rows	SUB420 Chrominance Rows
M rows, M odd	M rows	(M+1)/2 rows
M rows, M even	M rows	M/2 rows

43.6.18 Rounding, Limiting and Packing (RLP) Module

This module is used to round, limit and pack in the incoming pixel stream before the host DMA module. The RLP samples the sub420_data[39:0] 40-bit data bus and generates rlp_data[31:0] 32-bit data words with the associated validity signal rlp_valid.

Figure 43-35. RLP Block Diagram



ISC_RLP_CFG	RLP_DATA Slice	Value
DAT8	rlp_data[31:8]	0
	rlp_data[7:0]	sub420_data[11:4]
DAT9	rlp_data[31:9]	0
	rlp_data[8:0]	sub420_data[11:3]
DAT10	rlp_data[31:10]	0
	rlp_data[9:0]	sub420_data[11:2]
DAT11	rlp_data[31:11]	0
	rlp_data[10:0]	sub420_data[11:1]
DAT12	rlp_data[31:12]	0
	rlp_data[11:0]	sub420_data[11:0]
DATY8	rlp_data[31:8]	0
	rlp_data[7:0]	Y = rounded(sub420_data[29:22])
DATY10	rlp_data[31:8]	0
	rlp_data[7:0]	Y = sub420_data[29:20])
ARGB444	rlp_data[31:16]	0
	rlp_data[15:12]	A = alpha[7:4]
	rlp_data[11:8]	R = sub420_data[29:26]
	rlp_data[7:4]	G = sub420_data[19:16]
	rlp_data[3:0]	B = sub420_data[9:6]
ARGB555	rlp_data[31:16]	0
	rlp_data[15]	A = alpha[7]
	rlp_data[14:10]	R = sub420_data[29:25]
	rlp_data[9:5]	G = sub420_data[19:15]
	rlp_data[4:0]	B = sub420_data[9:5]
RGB565	rlp_data[31:16]	0
	rlp_data[15:11]	R = sub420_data[29:25]
	rlp_data[10:5]	G = sub420_data[19:14]
	rlp_data[4:0]	B = sub420_data[9:5]

...........continued

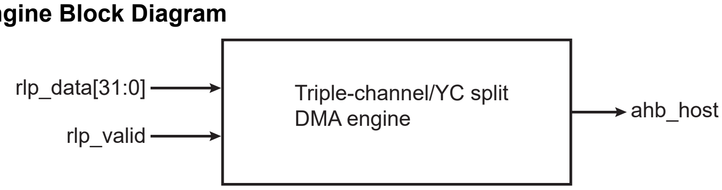
ISC_RLP_CFG	RLP_DATA Slice	Value
RGB32	rlp_data[31:24]	A = alpha[7:0]
	rlp_data[23:16]	R = sub420_data[29:22]
	rlp_data[15:8]	G = sub420_data[19:12]
	rlp_data[7:0]	B = sub420_data[9:2]
YCbCr422, YCbCr420	rlp_data[31:24]	Y1 = round(sub420_data[39:32])
	rlp_data[23:16]	Y0 = round(sub420_data[29:22])
	rlp_data[15:8]	Cb = round(sub420_data[19:12])
	rlp_data[7:0]	Cr = round(sub420_data[9:2])
YCbCr422, YCbCr420	rlp_data[31:24]	Y1 = round_limit(sub420_data[39:32])
	rlp_data[23:16]	Y0 = round_limit(sub420_data[29,22])
	rlp_data[15:8]	Cb = round_limit(sub420_data[19,12])
	rlp_data[7:0]	Cr = round_limit(sub420_data[9:2])
Undefined	rlp_data[31:0]	sub420_data[31:0]

ISC_RLP_CFG	8-bit Full Range	8-bit Limited Range
Y	0–255	16–235
Cb	0–255	16–240
Cr	0–255	16–240

43.6.19 DMA Interface

The descriptor-based DMA interface supports multiple buffers. A DMA stride value shows the offset between two consecutive lines (in bytes). If the stride is set to zero, the frame buffer is contiguous. When ISC_DCTRL.WB is set (Write Back), the DMA interface performs a single write operation to the ISC_DCTRL register, and sets ISC_DCTRL[7] to one and ISC_DCTRL[6] to the value of the frame field when interlaced content is being used. That means that interlaced fields are tagged with their relevant field values. The Write Back operation is always performed when the whole frame has been transferred to memory.

Figure 43-36. DMA Engine Block Diagram



ISC_DCFG.IMODE	DMA Engine Input Data
PACKED8	rlp_data[7:0]
PACKED16	rlp_data[15:0]
PACKED32	rlp_data[31:0]
YC422SP	rlp_data[31:0]
YC422P	rlp_data[31:0]
YC420SP	rlp_data[31:0]

...........continued

ISC_DCFG.IMODE	DMA Engine Input Data
YC420P	rlp_data[31:0]

When a bus error is detected, an interrupt flag is set. If the error occurs on a write operation, ISC_INTSR.WERR is asserted. If the error occurs on a read operation, ISC_INTSR.RERR is asserted. ISC_INTSR.WERRID gives details on the first error channel identifier.

43.6.19.1 Descriptor Memory Address Mapping

ISC_DCFG.IMODE	ISC_DAD0.AD0	ISC_DAD1.AD1	ISC_DAD2.AD2
PACKED8, PACKED16, PACKED32	data address	not used	not used
YC422SP	Y address	CbCr address	not used
YC422P	Y address	Cb address	Cr address
YC420SP	Y address	CbCr address	not used
YC420P	Y address	Cb address	Cr address

43.6.19.2 Descriptor Memory Mapping

Three descriptor views are available. Descriptor view 0 is used when the pixel or data stream is packed. Descriptor view 1 is used for YCbCr semi-planar pixel stream. Descriptor view 2 is used for YCbCr planar pixel stream.

Table 43-4. ISC_DCTRL.DVIEW = 0

Address	Register
ISC_DNDA+0x00	ISC_DCTRL
ISC_DNDA+0x04	ISC_DNDA
ISC_DNDA+0x08	ISC_DAD0
ISC_DNDA+0x0C	ISC_DST0

Table 43-5. ISC_DCTRL.DVIEW = 1

Address	Register
ISC_DNDA+0x00	ISC_DCTRL
ISC_DNDA+0x04	ISC_DNDA
ISC_DNDA+0x08	ISC_DAD0
ISC_DNDA+0x0C	ISC_DST0
ISC_DNDA+0x10	ISC_DAD1
ISC_DNDA+0x14	ISC_DST1

Table 43-6. ISC_DCTRL.DVIEW = 2

Address	Register
ISC_DNDA+0x00	ISC_DCTRL
ISC_DNDA+0x04	ISC_DNDA
ISC_DNDA+0x08	ISC_DAD0
ISC_DNDA+0x0C	ISC_DST0

...........continued	
Address	**Register**
ISC_DNDA+0x10	ISC_DAD1
ISC_DNDA+0x14	ISC_DST1
ISC_DNDA+0x18	ISC_DAD2
ISC_DNDA+0x1C	ISC_DST2

43.6.19.3 Example: Memory Mapping for 16-bit Packed, DMA Interface IMODE = 1 at ISC_DAD0.AD0 Location

Table 43-7. DAT8 Packing (ISC_RLP_CFG.MODE)

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW12	–	–	–	–	–	–	–	–	rlp_data1[7:0]								–	–	–	–	–	–	–	–	rlp_data0[7:0]							

Table 43-8. DAT9 Packing (ISC_RLP_CFG.MODE)

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW12	–	–	–	–	–	–	–	rlp_data1[8:0]									–	–	–	–	–	–	–	rlp_data0[8:0]								

Table 43-9. DAT10 Packing (ISC_RLP_CFG.MODE)

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW12	–	–	–	–	–	–	rlp_data1[9:0]										–	–	–	–	–	–	rlp_data0[9:0]									

Table 43-10. DAT11 Packing (ISC_RLP_CFG.MODE)

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW12	–	–	–	–	–	rlp_data1[10:0]											–	–	–	–	–	isc_data0[10:0]										

Table 43-11. DAT12 Packing (ISC_RLP_CFG.MODE)

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW12	–	–	–	–	rlp_data1[11:0]												–	–	–	–	rlp_data0[11:0]											

43.6.19.4 Example: Memory Mapping for 12-bit YC420SP, DMA Interface IMODE = 5

Table 43-12. Y Channel Located at ISC_DAD0.AD0 Memory Address

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y 8-bit	rlp_data1[31:24]								rlp_data1[23:16]								rlp_data0[31:24]								rlp_data0[23:16]							

Table 43-13. CbCr Channel Located at ISC_DAD1.AD1 Memory Address

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC 16-bit	rlp_data1[15:0]																rlp_data0[15:0]															

43.6.19.5 Example: Memory Mapping for 12-bit YC420P, DMA Interface IMODE = 6

Table 43-14. Y Channel Located at ISC_DAD0.AD0 Memory Address

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y 8-bit	rlp_data1[31:24]								rlp_data1[23:16]								rlp_data0[31:24]								rlp_data0[23:16]							

Table 43-15. Cb Channel Located at ISC_DAD1.AD1 Memory Address

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cb 8-bit	rlp_data3[15:8]							rlp_data2[15:8]									rlp_data1[15:8]							rlp_data0[15:8]								

Table 43-16. Cr Channel Located at ISC_DAD2.AD2

Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cr 8-bit	rlp_data3[7:0]						rlp_data2[7:0]										rlp_data1[7:0]						rlp_data0[7:0]									

43.6.20 Histogram Module

For each possible pixel value, the histogram counts the number of times the value was encountered in the current image. RGGB Bayer, raw data or luminance histogram are available. There are 512 entries in the histogram entries, and each histogram bin can count up to 2^20 data. As the table entries are limited, each bin is actually a range, i.e., least significant bits are ignored. A write to ISC_CTRLEN.HISREQ initiates a new histogram. The counting operation ends when ISC_INTSR.HISDONE is set. At that time, a software or hardware dma transfer copies the table from the interface to the internal or external memory. To clear the table content (for a new operation), use ISC_CTRLEN.HISCLR. An automatic clear (reset after read) is available when ISC_HIS_CFG.RAR is set. In that case, as soon as the data is read from the table, the table entry is cleared.

Figure 43-37. Histogram Block Diagram

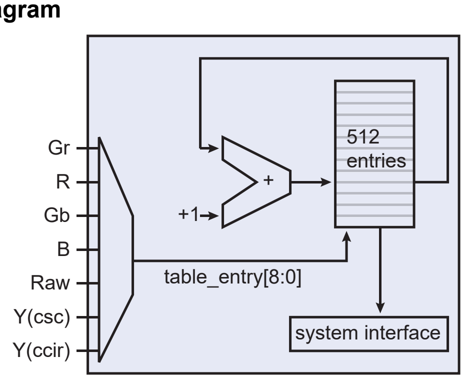
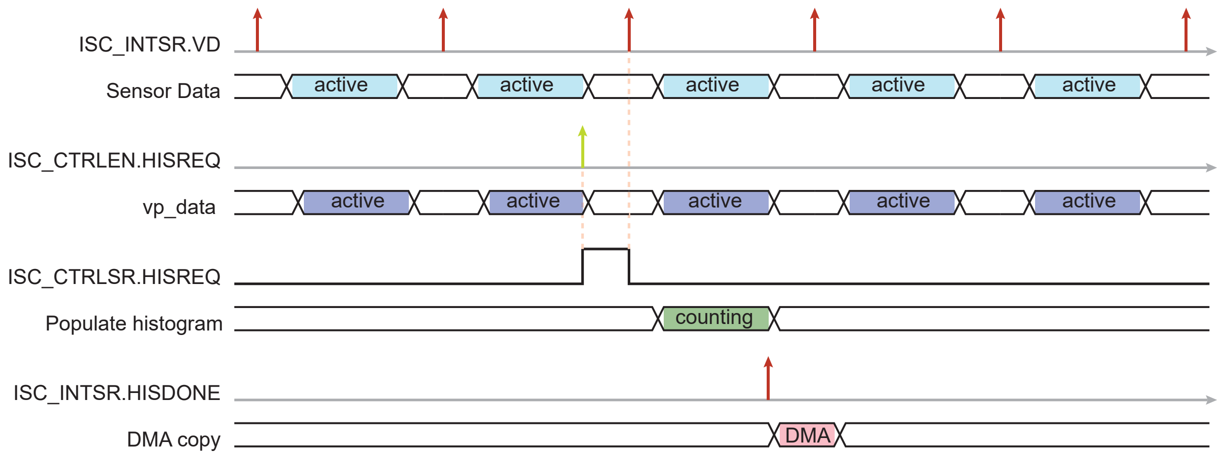


Figure 43-38. Histogram Request Timing Diagram



43.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ISC_CTRLEN	31:24								
		23:16								
		15:8							FUPPRO	
		7:0					HISCLR	HISREQ	UPPRO	CAPTURE
0x04	ISC_CTRLDIS	31:24								
		23:16								
		15:8								SWRST
		7:0								DISABLE
0x08	ISC_CTRLSR	31:24	SIP							
		23:16								
		15:8								
		7:0				FIELD		HISREQ	UPPRO	CAPTURE
0x0C	ISC_PFE_CFG0	31:24	REP	BPS[2:0]			CCIR_REP			
		23:16	SKIPCNT[7:0]							
		15:8		MIPI	ROWEN	COLEN	CCIR10_8N	CCIR_CRC	CCIR656	GATED
		7:0	CONT	MODE[2:0]			FPOL	PPOL	VPOL	HPOL
0x10	ISC_PFE_CFG1	31:24	COLMAX[15:8]							
		23:16	COLMAX[7:0]							
		15:8	COLMIN[15:8]							
		7:0	COLMIN[7:0]							
0x14	ISC_PFE_CFG2	31:24	ROWMAX[15:8]							
		23:16	ROWMAX[7:0]							
		15:8	ROWMIN[15:8]							
		7:0	ROWMIN[7:0]							
0x18	ISC_CLKEN	31:24								
		23:16								
		15:8								
		7:0							MCEN	ICEN
0x1C	ISC_CLKDIS	31:24								
		23:16								
		15:8							MCSWRST	ICSWRST
		7:0							MCDIS	ICDIS
0x20	ISC_CLKSR	31:24	SIP							
		23:16								
		15:8								
		7:0							MCSR	ICSR
0x24	ISC_CLKCFG	31:24								
		23:16	MCDIV[7:0]							
		15:8								
		7:0								
0x28	ISC_INTEN	31:24			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
		23:16				RERR				WERR
		15:8			HISCLR	HISDONE			LDONE	DDONE
		7:0			DIS	SWRST			HD	VD
0x2C	ISC_INTDIS	31:24			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
		23:16				RERR				WERR
		15:8			HISCLR	HISDONE			LDONE	DDONE
		7:0			DIS	SWRST			HD	VD
0x30	ISC_INTMASK	31:24			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
		23:16				RERR				WERR
		15:8			HISCLR	HISDONE			LDONE	DDONE
		7:0			DIS	SWRST			HD	VD
0x34	ISC_INTSR	31:24			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
		23:16				RERR		WERRID[1:0]		WERR
		15:8			HISCLR	HISDONE			LDONE	DDONE
		7:0			DIS	SWRST			HD	VD

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38 ... 0x3F	Reserved									
0x40	ISC_DPC_CTRL	31:24								
		23:16								
		15:8								
		7:0						BLCEN	GDCEN	DPCEN
0x44	ISC_DPC_CFG	31:24	BLOFST[8:1]							
		23:16	BLOFST[0]	GDCCLP[2:0]					RE_MODE	ND_MODE
		15:8		TA_ENABLE	TC_ENABLE	TM_ENABLE				
		7:0				EITPOL			BAYCFG[1:0]	
0x48	ISC_DPC_THRESHM	31:24								
		23:16								
		15:8					THRESHM[11:8]			
		7:0	THRESHM[7:0]							
0x4C	ISC_DPC_THRESHC	31:24								
		23:16								
		15:8					THRESHC[11:8]			
		7:0	THRESHC[7:0]							
0x50	ISC_DPC_THRESHA	31:24								
		23:16								
		15:8					THRESHA[11:8]			
		7:0	THRESHA[7:0]							
0x54	ISC_DPC_SR	31:24								
		23:16	COUNTER[23:16]							
		15:8	COUNTER[15:8]							
		7:0	COUNTER[7:0]							
0x58	ISC_WB_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x5C	ISC_WB_CFG	31:24								
		23:16								
		15:8								
		7:0							BAYCFG[1:0]	
0x60	ISC_WB_O_RGR	31:24				GROFST[12:8]				
		23:16	GROFST[7:0]							
		15:8				ROFST[12:8]				
		7:0	ROFST[7:0]							
0x64	ISC_WB_O_BGB	31:24				GBOFST[12:8]				
		23:16	GBOFST[7:0]							
		15:8				BOFST[12:8]				
		7:0	BOFST[7:0]							
0x68	ISC_WB_G_RGR	31:24				GRGAIN[12:8]				
		23:16	GRGAIN[7:0]							
		15:8				RGAIN[12:8]				
		7:0	RGAIN[7:0]							
0x6C	ISC_WB_G_BGB	31:24				GBGAIN[12:8]				
		23:16	GBGAIN[7:0]							
		15:8				BGAIN[12:8]				
		7:0	BGAIN[7:0]							
0x70	ISC_CFA_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x74	ISC_CFA_CFG	31:24								
		23:16								
		15:8					EAL[1:0]			
		7:0				EITPOL			BAYCFG[1:0]	

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	ISC_CC_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x7C	ISC_CC_RR_RG	31:24					RGGAIN[11:8]			
		23:16	RGGAIN[7:0]							
		15:8					RRGAIN[11:8]			
		7:0	RRGAIN[7:0]							
0x80	ISC_CC_RB_OR	31:24				ROFST[12:8]				
		23:16	ROFST[7:0]							
		15:8					RBGAIN[11:8]			
		7:0	RBGAIN[7:0]							
0x84	ISC_CC_GR_GG	31:24					GGGAIN[11:8]			
		23:16	GGGAIN[7:0]							
		15:8					GRGAIN[11:8]			
		7:0	GRGAIN[7:0]							
0x88	ISC_CC_GB_OG	31:24				GOFST[12:8]				
		23:16	GOFST[7:0]							
		15:8					GBGAIN[11:8]			
		7:0	GBGAIN[7:0]							
0x8C	ISC_CC_BR_BG	31:24					BGGAIN[11:8]			
		23:16	BGGAIN[7:0]							
		15:8					BRGAIN[11:8]			
		7:0	BRGAIN[7:0]							
0x90	ISC_CC_BB_OB	31:24				BOFST[12:8]				
		23:16	BOFST[7:0]							
		15:8					BBGAIN[11:8]			
		7:0	BBGAIN[7:0]							
0x94	ISC_GAM_CTRL	31:24								
		23:16								
		15:8								
		7:0				BIPART	RENABLE	GENABLE	BENABLE	ENABLE
0x98	ISC_GAM_BENTRY0	31:24							BCONSTANT[9:8]	
		23:16	BCONSTANT[7:0]							
		15:8							BSLOPE[9:8]	
		7:0	BSLOPE[7:0]							
...										
0x0194	ISC_GAM_BENTRY63	31:24							BCONSTANT[9:8]	
		23:16	BCONSTANT[7:0]							
		15:8							BSLOPE[9:8]	
		7:0	BSLOPE[7:0]							
0x0198	ISC_GAM_GENTRY0	31:24							GCONSTANT[9:8]	
		23:16	GCONSTANT[7:0]							
		15:8							GSLOPE[9:8]	
		7:0	GSLOPE[7:0]							
...										
0x0294	ISC_GAM_GENTRY63	31:24							GCONSTANT[9:8]	
		23:16	GCONSTANT[7:0]							
		15:8							GSLOPE[9:8]	
		7:0	GSLOPE[7:0]							
0x0298	ISC_GAM_RENTRY0	31:24							RCONSTANT[9:8]	
		23:16	RCONSTANT[7:0]							
		15:8							RSLOPE[9:8]	
		7:0	RSLOPE[7:0]							
...										
0x0394	ISC_GAM_RENTRY63	31:24							RCONSTANT[9:8]	
		23:16	RCONSTANT[7:0]							
		15:8							RSLOPE[9:8]	
		7:0	RSLOPE[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0398	ISC_VHXS_CTRL	31:24								
		23:16								
		15:8								
		7:0							HXSEN	VXSEN
0x039C	ISC_VHXS_SS	31:24					YS[11:8]			
		23:16	YS[7:0]							
		15:8					XS[11:8]			
		7:0	XS[7:0]							
0x03A0	ISC_VHXS_DS	31:24					YD[11:8]			
		23:16	YD[7:0]							
		15:8					XD[11:8]			
		7:0	XD[7:0]							
0x03A4	ISC_VXS_FACT	31:24								
		23:16	VFACT[23:16]							
		15:8	VFACT[15:8]							
		7:0	VFACT[7:0]							
0x03A8	ISC_HXS_FACT	31:24								
		23:16	HFACT[23:16]							
		15:8	HFACT[15:8]							
		7:0	HFACT[7:0]							
0x03AC	ISC_VXS_CFG	31:24	FLMAX[3:0]				FLMIN[3:0]			
		23:16								
		15:8					OFFSET[3:0]			
		7:0				TAP2			FILTCFG[1:0]	
0x03B0	ISC_HXS_CFG	31:24					FL[3:0]			
		23:16								
		15:8					OFFSET[3:0]			
		7:0				TAP2			FILTCFG[1:0]	
0x03B4	ISC_VXS_TAP10PHI0	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03B8	ISC_VXS_TAP32PHI0	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03BC	ISC_VXS_TAP10PHI1	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03C0	ISC_VXS_TAP32PHI1	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03C4	ISC_VXS_TAP10PHI2	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03C8	ISC_VXS_TAP32PHI2	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03CC	ISC_VXS_TAP10PHI3	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03D0	ISC_VXS_TAP32PHI3	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03D4	ISC_VXS_TAP10PHI4	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03D8	ISC_VXS_TAP32PHI4	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03DC	ISC_VXS_TAP10PHI5	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03E0	ISC_VXS_TAP32PHI5	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03E4	ISC_VXS_TAP10PHI6	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03E8	ISC_VXS_TAP32PHI6	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03EC	ISC_VXS_TAP10PHI7	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03F0	ISC_VXS_TAP32PHI7	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03F4	ISC_VXS_TAP10PHI8	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x03F8	ISC_VXS_TAP32PHI8	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x03FC	ISC_VXS_TAP10PHI9	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0400	ISC_VXS_TAP32PHI9	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0404	ISC_VXS_TAP10PHI10	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0408	ISC_VXS_TAP32PHI10	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x040C	ISC_VXS_TAP10PHI11	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0410	ISC_VXS_TAP32PHI11	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0414	ISC_VXS_TAP10PHI12	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0418	ISC_VXS_TAP32PHI12	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x041C	ISC_VXS_TAP10PHI13	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0420	ISC_VXS_TAP32PHI13	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0424	ISC_VXS_TAP10PHI14	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0428	ISC_VXS_TAP32PHI14	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x042C	ISC_VXS_TAP10PHI15	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0430	ISC_VXS_TAP32PHI15	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0434	ISC_HXS_TAP10PHI0	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0438	ISC_HXS_TAP32PHI0	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x043C	ISC_HXS_TAP10PHI1	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0440	ISC_HXS_TAP32PHI1	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0444	ISC_HXS_TAP10PHI2	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0448	ISC_HXS_TAP32PHI2	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x044C	ISC_HXS_TAP10PHI3	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0450	ISC_HXS_TAP32PHI3	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0454	ISC_HXS_TAP10PHI4	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0458	ISC_HXS_TAP32PHI4	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x045C	ISC_HXS_TAP10PHI5	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0460	ISC_HXS_TAP32PHI5	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0464	ISC_HXS_TAP10PHI6	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0468	ISC_HXS_TAP32PHI6	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x046C	ISC_HXS_TAP10PHI7	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0470	ISC_HXS_TAP32PHI7	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0474	ISC_HXS_TAP10PHI8	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0478	ISC_HXS_TAP32PHI8	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x047C	ISC_HXS_TAP10PHI9	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0480	ISC_HXS_TAP32PHI9	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0484	ISC_HXS_TAP10PHI10	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0488	ISC_HXS_TAP32PHI10	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x048C	ISC_HXS_TAP10PHI11	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0490	ISC_HXS_TAP32PHI11	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0494	ISC_HXS_TAP10PHI12	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0498	ISC_HXS_TAP32PHI12	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x049C	ISC_HXS_TAP10PHI13	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x04A0	ISC_HXS_TAP32PHI13	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x04A4	ISC_HXS_TAP10PHI14	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x04A8	ISC_HXS_TAP32PHI14	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x04AC	ISC_HXS_TAP10PHI15	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04B0	ISC_HXS_TAP32PHI15	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x04B4	ISC_CSC_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x04B8	ISC_CSC_YR_YG	31:24					YGGAIN[11:8]			
		23:16	YGGAIN[7:0]							
		15:8					YRGAIN[11:8]			
		7:0	YRGAIN[7:0]							
0x04BC	ISC_CSC_YB_OY	31:24						YOFST[10:8]		
		23:16	YOFST[7:0]							
		15:8					YBGAIN[11:8]			
		7:0	YBGAIN[7:0]							
0x04C0	ISC_CSC_CBR_CBG	31:24					CBGGAIN[11:8]			
		23:16	CBGGAIN[7:0]							
		15:8					CBRGAIN[11:8]			
		7:0	CBRGAIN[7:0]							
0x04C4	ISC_CSC_CBB_OCB	31:24						CBOFST[10:8]		
		23:16	CBOFST[7:0]							
		15:8					CBBGAIN[11:8]			
		7:0	CBBGAIN[7:0]							
0x04C8	ISC_CSC_CRR_CRG	31:24					CRGGAIN[11:8]			
		23:16	CRGGAIN[7:0]							
		15:8					CRRGAIN[11:8]			
		7:0	CRRGAIN[7:0]							
0x04CC	ISC_CSC_CRB_OCR	31:24						CROFST[10:8]		
		23:16	CROFST[7:0]							
		15:8					CRBGAIN[11:8]			
		7:0	CRBGAIN[7:0]							
0x04D0	ISC_CBHS_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x04D4	ISC_CBHS_CFG	31:24								
		23:16								
		15:8								
		7:0						CCIRMODE[1:0]		CCIR
0x04D8	ISC_CBHS_BRIGHT	31:24								
		23:16								
		15:8						BRIGHT[10:8]		
		7:0	BRIGHT[7:0]							
0x04DC	ISC_CBHS_CONTRAST	31:24								
		23:16								
		15:8					CONTRAST[11:8]			
		7:0	CONTRAST[7:0]							
0x04E0	ISC_CBHS_HUE	31:24								
		23:16								
		15:8								HUE[8]
		7:0	HUE[7:0]							
0x04E4	ISC_CBHS_SAT	31:24								
		23:16								
		15:8					SATURATION[11:8]			
		7:0	SATURATION[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04E8	ISC_SUB422_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x04EC	ISC_SUB422_CFG	31:24								
		23:16								
		15:8								
		7:0			FILTER[1:0]			CCIRMODE[1:0]		CCIR
0x04F0	ISC_SUB420_CTRL	31:24								
		23:16								
		15:8								
		7:0			MIPI420	FILTER				ENABLE
0x04F4	ISC_RLP_CFG	31:24								
		23:16								
		15:8	ALPHA[7:0]							
		7:0	YMODE[1:0]		LSH	REP	MODE[3:0]			
0x04F8	ISC_HIS_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x04FC	ISC_HIS_CFG	31:24								
		23:16								
		15:8								RAR
		7:0			BAYSEL[1:0]			MODE[2:0]		
0x0500 ... 0x051B	Reserved									
0x051C	ISC_DCFG	31:24								
		23:16	AWQOS[3:0]				ARQOS[3:0]			
		15:8						CMBSIZE[2:0]		
		7:0		YMBSIZE[2:0]				IMODE[2:0]		
0x0520	ISC_DCTRL	31:24								
		23:16								
		15:8								
		7:0	DONE	FIELD	WB	IE		DVIEW[1:0]		DE
0x0524	ISC_DNDA	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							
0x0528	ISC_DAD0	31:24	AD0[31:24]							
		23:16	AD0[23:16]							
		15:8	AD0[15:8]							
		7:0	AD0[7:0]							
0x052C	ISC_DST0	31:24	ST0[31:24]							
		23:16	ST0[23:16]							
		15:8	ST0[15:8]							
		7:0	ST0[7:0]							
0x0530	ISC_DAD1	31:24	AD1[31:24]							
		23:16	AD1[23:16]							
		15:8	AD1[15:8]							
		7:0	AD1[7:0]							
0x0534	ISC_DST1	31:24	ST1[31:24]							
		23:16	ST1[23:16]							
		15:8	ST1[15:8]							
		7:0	ST1[7:0]							
0x0538	ISC_DAD2	31:24	AD2[31:24]							
		23:16	AD2[23:16]							
		15:8	AD2[15:8]							
		7:0	AD2[7:0]							

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x053C	ISC_DST2	31:24	ST2[31:24]							
		23:16	ST2[23:16]							
		15:8	ST2[15:8]							
		7:0	ST2[7:0]							
0x0540 ... 0x055B	Reserved									
0x055C	ISC_HIS_ENTRY0	31:24								
		23:16					COUNT[19:16]			
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
...										
0x0D58	ISC_HIS_ENTRY511	31:24								
		23:16					COUNT[19:16]			
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							

43.7.1 ISC Control Enable Register

Name: ISC_CTRLEN
Offset: 0x00
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding command.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							FUPPRO	
Access							W	
Reset							–	
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					HISCLR	HISREQ	UPPRO	CAPTURE
Access					W	W	W	W
Reset					–	–	–	–

Bit 9 – FUPPRO Force Update Color Profile

Bit 3 – HISCLR Histogram Clear

Bit 2 – HISREQ Histogram Request

Bit 1 – UPPRO Update Profile

Bit 0 – CAPTURE Capture Input Stream Command

43.7.2 ISC Control Disable Register

Name: ISC_CTRLDIS
Offset: 0x04
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Performs the corresponding command.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DISABLE
Access								W
Reset								–

Bit 8 – SWRST Software Reset

Bit 0 – DISABLE Capture Disable

43.7.3 ISC Control Status Register

Name: ISC_CTRLSR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SIP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				FIELD		HISREQ	UPPRO	CAPTURE
Access				R		R	R	R
Reset				0		0	0	0

Bit 31 – SIP Synchronization In Progress

Value	Description
0	The double domain synchronization is terminated.
1	The double domain synchronization is in progress.

Bit 4 – FIELD Field Status (only relevant when the video stream is interlaced)

Value	Description
0	The current field/segment is a top field
1	The current field/segment is a bottom field.

Bit 2 – HISREQ Histogram Request Pending

Value	Description
0	There is no histogram pending request.
1	Indicates that the histogram request is still pending.

Bit 1 – UPPRO Profile Update Pending

Value	Description
0	There is no profile update pending request.
1	Indicates that the profile update request is still pending.

Bit 0 – CAPTURE Capture Pending

Value	Description
0	Capture mode is disabled.
1	Capture is pending.

43.7.4 ISC Parallel Front End Configuration 0 Register

Name: ISC_PFE_CFG0
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	REP	BPS[2:0]			CCIR_REP			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	23	22	21	20	19	18	17	16
	SKIPCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIPI	ROWEN	COLEN	CCIR10_8N	CCIR_CRC	CCIR656	GATED
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONT	MODE[2:0]			FPOL	PPOL	VPOL	HPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – REP Up Multiply with Replication

Value	Description
0	Unused bits are stuck at 0.
1	Unused bits are copied from MSB.

Bits 30:28 – BPS[2:0] Bits Per Sample

Value	Name	Description
0	TWELVE	12-bit input
1	ELEVEN	11-bit input
2	TEN	10-bit input
3	NINE	9-bit input
4	EIGHT	8-bit input
5	FORTY	40-bit input (used for MIPI formats up to forty bits per pixel)

Bit 27 – CCIR_REP CCIR Replication

Value	Description
0	Unused bits are stuck at 0.
1	Unused bits are copied from MSB.

Bits 23:16 – SKIPCNT[7:0] Frame Skipping Counter

Bit 14 – MIPI MIPI Interface Connection

Value	Description
0	Input data are coming from the physical parallel interface.
1	Input data are coming from the physical MIPI interface.

Bit 13 – ROWEN Row Cropping Enable

Value	Description
0	Row Cropping is disabled.
1	Row Cropping is enabled.

Bit 12 – COLEN Column Cropping Enable

Value	Description
0	Column Cropping is disabled.
1	Column Cropping is enabled.

Bit 11 – CCIR10_8N CCIR 10 bits or 8 bits

Value	Description
0	8-bit mode.
1	10-bit mode.

Bit 10 – CCIR_CRC CCIR656 CRC Decoder

Value	Description
0	Embedded CRC is discarded.
1	Embedded CRC is decoded.

Bit 9 – CCIR656 CCIR656 input mode

Value	Description
0	HSYNC and VSYNC signals are used to synchronize the input stream.
1	Embedded synchronization is used.

Bit 8 – GATED Gated input clock

Value	Description
0	The external pixel clock is free running.
1	The external pixel clock is gated.

Bit 7 – CONT Continuous Acquisition

Value	Description
0	Single Shot mode.
1	Video mode.

Bits 6:4 – MODE[2:0] Parallel Front End Mode

Value	Name	Description
0	PROGRESSIVE	Video source is progressive.
1	DF_TOP	Video source is interlaced, two fields are captured starting with top field.
2	DF_BOTTOM	Video source is interlaced, two fields are captured starting with bottom field.
3	DF_IMMEDIATE	Video source is interlaced, two fields are captured immediately.
4	SF_TOP	Video source is interlaced, one field is captured starting with the top field.
5	SF_BOTTOM	Video source is interlaced, one field is captured starting with the bottom field.
6	SF_IMMEDIATE	Video source is interlaced, one field is captured starting immediately.

Bit 3 – FPOL Field Polarity

Value	Description
0	Top field is sampled when F value is 0; Bottom field is sampled when F value is 1.
1	Top field is sampled when F value is 1; Bottom field is sampled when F value is 0.

Bit 2 – PPOL Pixel Clock Polarity

Value	Description
0	The pixel stream is sampled on the rising edge of the pixel clock.
1	The pixel stream is sampled on the falling edge of the pixel clock.

Bit 1 – VPOL Vertical Synchronization Polarity

Value	Description
0	VSYNC signal is active high, i.e. valid pixels are sampled when VSYNC is asserted.
1	VSYNC signal is active low, i.e. valid pixels are sampled when VSYNC is deasserted.

Bit 0 – HPOL Horizontal Synchronization Polarity

Value	Description
0	HSYNC signal is active high, i.e. valid pixels are sampled when HSYNC is asserted.
1	HSYNC signal is active low, i.e. valid pixels are sampled when HSYNC is deasserted.

43.7.5 ISC Parallel Front End Configuration 1 Register

Name: ISC_PFE_CFG1
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	COLMAX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COLMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COLMIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COLMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – COLMAX[15:0] Column Maximum Limit
Horizontal ending position of the cropping area.

Bits 15:0 – COLMIN[15:0] Column Minimum Limit
Horizontal starting position of the cropping area.

43.7.6 ISC Parallel Front End Configuration 2 Register

Name: ISC_PFE_CFG2
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ROWMAX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ROWMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ROWMIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ROWMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ROWMAX[15:0] Row Maximum Limit
Vertical ending position of the cropping area.

Bits 15:0 – ROWMIN[15:0] Row Minimum Limit
Vertical starting position of the cropping area.

43.7.7 ISC Clock Enable Register

Name: ISC_CLKEN
Offset: 0x18
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							MCEN	ICEN
Access							W	W
Reset							–	–

Bit 1 – MCEN Camera Sensor Clock Domain Enable

Value	Description
0	No effect.
1	Enables the camera sensor clock.

Bit 0 – ICEN ISP Clock Enable

Value	Description
0	No effect.
1	Enables the ISP clock.

43.7.8 ISC Clock Disable Register

Name: ISC_CLKDIS
Offset: 0x1C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Performs the corresponding command.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							MCSWRST	ICSWRST
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
							MCDIS	ICDIS
Access							W	W
Reset							–	–

Bit 9 – MCSWRST Camera Sensor Clock Domain Software Reset

Bit 8 – ICSWRST ISP Clock Software Reset

Bit 1 – MCDIS Camera Sensor Clock Domain Disable

Bit 0 – ICDIS ISP Clock Disable

43.7.9 ISC Clock Status Register

Name: ISC_CLKSR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SIP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							MCSR	ICSR
Access							R	R
Reset							0	0

Bit 31 – SIP Synchronization In Progress

Value	Description
0	The double domain synchronization operation is over.
1	The double domain synchronization operation is in progress.

Bit 1 – MCSR Camera Sensor Clock Status Register

Value	Description
0	The camera sensor clock is disabled.
1	The camera sensor clock is enabled.

Bit 0 – ICSR ISP Clock Status Register

Value	Description
0	The ISP clock is disabled.
1	The ISP clock is enabled.

43.7.10 ISC Clock Configuration Register

Name: ISC_CLKCFG
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	MCDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Access								
Reset								

Bits 23:16 – MCDIV[7:0] Camera Sensor Reference Clock Divider

$$f_{\mathrm{mc}} = \frac{f_{\mathrm{mcref}}}{\mathrm{MCDIV} + 1}$$

43.7.11 ISC Interrupt Enable Register

Name: ISC_INTEN
Offset: 0x28
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the interrupt.

Bit	31	30	29	28	27	26	25	24
			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
				RERR				WERR
Access				W				W
Reset				–				–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
			HISCLR	HISDONE			LDONE	DDONE
Access			W	W			W	W
Reset			–	–			–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			DIS	SWRST			HD	VD
Access			W	W			W	W
Reset			–	–			–	–

Bit 29 – GFOV Input FIFO Overflow Interrupt Enable

Bit 28 – CCIRERR CCIR Decoder Error Interrupt Enable

Bit 27 – HDTO Horizontal Synchronization Timeout Interrupt Enable

Bit 26 – VDTO Vertical Synchronization Timeout Interrupt Enable

Bit 25 – DAOV Data Overflow Interrupt Enable

Bit 24 – VFPOV Vertical Front Porch Overflow Interrupt Enable

Bit 20 – RERR Read Channel Error Interrupt Enable

Bit 16 – WERR Write Channel Error Interrupt Enable

Bit 13 – HISCLR Histogram Clear Interrupt Enable

Bit 12 – HISDONE Histogram Completed Interrupt Enable

Bit 9 – LDONE DMA List Done Interrupt Enable

Bit 8 – DDONE DMA Done Interrupt Enable

Bit 5 – DIS Disable Completed Interrupt Enable

Bit 4 – SWRST Software Reset Completed Interrupt Enable

Bit 1 – HD Horizontal Synchronization Detection Interrupt Enable

Bit 0 – VD Vertical Synchronization Detection Interrupt Enable

43.7.12 ISC Interrupt Disable Register

Name: ISC_INTDIS
Offset: 0x2C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the interrupt.

Bit	31	30	29	28	27	26	25	24
			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
				RERR				WERR
Access				W				W
Reset				–				–

Bit	15	14	13	12	11	10	9	8
			HISCLR	HISDONE			LDONE	DDONE
Access			W	W			W	W
Reset			–	–			–	–

Bit	7	6	5	4	3	2	1	0
			DIS	SWRST			HD	VD
Access			W	W			W	W
Reset			–	–			–	–

Bit 29 – GFOV FIFO Overflow Interrupt Disable

Bit 28 – CCIRERR CCIR Decoder Error Interrupt Disable

Bit 27 – HDTO Horizontal Synchronization Timeout Interrupt Disable

Bit 26 – VDTO Vertical Synchronization Timeout Interrupt Disable

Bit 25 – DAOV Data Overflow Interrupt Disable

Bit 24 – VFPOV Vertical Front Porch Overflow Interrupt Disable

Bit 20 – RERR Read Channel Error Interrupt Disable

Bit 16 – WERR Write Channel Error Interrupt Disable

Bit 13 – HISCLR Histogram Clear Interrupt Disable

Bit 12 – HISDONE Histogram Completed Interrupt Disable

Bit 9 – LDONE DMA List Done Interrupt Disable

Bit 8 – DDONE DMA Done Interrupt Disable

Bit 5 – DIS Disable Completed Interrupt Disable

Bit 4 – SWRST Software Reset Completed Interrupt Disable

Bit 1 – HD Horizontal Synchronization Detection Interrupt Disable

Bit 0 – VD Vertical Synchronization Detection Interrupt Disable

43.7.13 ISC Interrupt Mask Register

Name: ISC_INTMASK
Offset: 0x30
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The interrupt is disabled.

1: The interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
Access			W	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				RERR				WERR
Access				R				R
Reset				0				0

Bit	15	14	13	12	11	10	9	8
			HISCLR	HISDONE			LDONE	DDONE
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			DIS	SWRST			HD	VD
Access			R	R			R	R
Reset			0	0			0	0

Bit 29 – GFOV FIFO Overflow Interrupt Mask

Bit 28 – CCIRERR CCIR Decoder Error Interrupt Mask

Bit 27 – HDTO Horizontal Synchronization Timeout Interrupt Mask

Bit 26 – VDTO Vertical Synchronization Timeout Interrupt Mask

Bit 25 – DAOV Data Overflow Interrupt Mask

Bit 24 – VFPOV Vertical Front Porch Overflow Interrupt Mask

Bit 20 – RERR Read Channel Error Interrupt Mask

Bit 16 – WERR Write Channel Error Interrupt Mask

Bit 13 – HISCLR Histogram Clear Interrupt Mask

Bit 12 – HISDONE Histogram Completed Interrupt Mask

Bit 9 – LDONE DMA List Done Interrupt Mask

Bit 8 – DDONE DMA Done Interrupt Mask

Bit 5 – DIS Disable Completed Interrupt Mask

Bit 4 – SWRST Software Reset Completed Interrupt Mask

Bit 1 – HD Horizontal Synchronization Detection Interrupt Mask

Bit 0 – VD Vertical Synchronization Detection Interrupt Mask

43.7.14 ISC Interrupt Status Register

Name: ISC_INTSR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			GFOV	CCIRERR	HDTO	VDTO	DAOV	VFPOV
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				RERR		WERRID[1:0]		WERR
Access				R		R	R	R
Reset				0		0	0	0

Bit	15	14	13	12	11	10	9	8
			HISCLR	HISDONE			LDONE	DDONE
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			DIS	SWRST			HD	VD
Access			R	R			R	R
Reset			0	0			0	0

Bit 29 – GFOV FIFO Overflow Interrupt (relevant if MIPI interface is not selected) (cleared on read)

Value	Description
0	No FIFO overflow detected since the last read of the Interrupt Status register.
1	A FIFO overflow has been detected.

Bit 28 – CCIRERR CCIR Decoder Error Interrupt (cleared on read)

Value	Description
0	No CCIR CRC error detected since the last read of the Interrupt Status register.
1	A CCIR CRC error has been detected.

Bit 27 – HDTO Horizontal Synchronization Timeout Interrupt (cleared on read)

Value	Description
0	A horizontal synchronization is detected.
1	No horizontal synchronization is detected.

Bit 26 – VDTO Vertical Synchronization Timeout Interrupt (cleared on read)

Value	Description
0	A vertical synchronization is detected.
1	No vertical synchronization is detected.

Bit 25 – DAOV Data Overflow Interrupt (cleared on read)

Value	Description
0	No data overflow error occurred since the last reset of the Interrupt Status register.
1	A data overflow occurred.

Bit 24 – VFPOV Vertical Front Porch Overflow Interrupt (cleared on read)

Value	Description
0	No vertical front porch error occurred since the last read of the Interrupt Status register.

Value	Description
1	The vertical synchronization has been detected but the DMA channel is still busy.

Bit 20 – RERR Read Channel Error Interrupt (cleared on read)

Value	Description
0	No read channel error since the last read of the Interrupt Status register.
1	A read channel error occurred when the ISC read the descriptor.

Bits 18:17 – WERRID[1:0] Write Channel Error Identifier (cleared on read)

Value	Name	Description
0	CH0	An error occurred for Channel 0 (RAW/RGB/Y)
1	CH1	An error occurred for Channel 1 (CbCr/Cb)
2	CH2	An error occurred for Channel 2 (Cr)
3	WB	Write back channel error

Bit 16 – WERR Write Channel Error Interrupt (cleared on read)

Value	Description
0	No write channel error since the last read of the Interrupt Status register.
1	A write channel error occurred.

Bit 13 – HISCLR Histogram Clear Interrupt (cleared on read)

Value	Description
0	No Histogram Clear Interrupt has been raised since the last read of the Interrupt Status register.
1	The Histogram Clear Interrupt has occurred.

Bit 12 – HISDONE Histogram Completed Interrupt (cleared on read)

Value	Description
0	No Histogram Completed Interrupt has been raised since the last read of the Interrupt Status register.
1	The Histogram Completed Interrupt has occurred.

Bit 9 – LDONE DMA List Done Interrupt (cleared on read)

Value	Description
0	No DMA List Done Interrupt has occurred since the last read of the Interrupt Status register.
1	The DMA List Done Interrupt has occurred.

Bit 8 – DDONE DMA Done Interrupt (cleared on read)

Value	Description
0	No DMA Transfer Done Interrupt has occurred since the last read of the Interrupt Status register.
1	The DMA Transfer Done Interrupt has occurred.

Bit 5 – DIS Disable Completed Interrupt (cleared on read)

Value	Description
0	The disable has not occurred since the last read of the Interrupt Status register.
1	The disable has completed.

Bit 4 – SWRST Software Reset Completed Interrupt (cleared on read)

Value	Description
0	No software reset completion since the last read of the Interrupt Status register.
1	The software reset has completed.

Bit 1 – HD Horizontal Synchronization Detected Interrupt (cleared on read)

Value	Description
0	No horizontal synchronization detection since the last read of the Interrupt Status register.
1	A horizontal synchronization has been detected.

Bit 0 – VD Vertical Synchronization Detected Interrupt (cleared on read)

Value	Description
0	No vertical synchronization detection since the last read of the Interrupt Status register.
1	A vertical synchronization has been detected.

43.7.15 ISC Defective Pixel Control Register

Name: ISC_DPC_CTRL
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						BLCEN	GDCEN	DPCEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – BLCEN Black Level Correction Enable

Value	Description
0	Black level correction is disabled.
1	Black level correction is enabled.

Bit 1 – GDCEN Green Disparity Correction Enable

Value	Description
0	Green disparity correction is disabled.
1	Green disparity correction is enabled.

Bit 0 – DPCEN Defective Pixel Correction Enable

Value	Description
0	Defective pixel correction is disabled.
1	Defective pixel correction is enabled.

43.7.16 ISC Defective Pixel Configuration Register

Name: ISC_DPC_CFG
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BLOFST[8:1]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLOFST[0]	GDCCLP[2:0]					RE_MODE	ND_MODE
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
		TA_ENABLE	TC_ENABLE	TM_ENABLE				
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bit	7	6	5	4	3	2	1	0
				EITPOL			BAYCFG[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 31:23 – BLOFST[8:0] Black Level Offset Value
BLOFST is the default constant value subtracted from the incoming stream. The value is unsigned, 0:8:0 value.

Bits 22:20 – GDCCLP[2:0] Green Disparity Clipping Value
Green Disparity Clipping is performed between [$-2^{GDCCLP+1}$, $-2^{GDCCLP+1}-1$]

Bit 17 – RE_MODE Replacement Algorithm

Value	Description
0	Median pixel is used.
1	Average pixel is used.

Bit 16 – ND_MODE Noise Detection Mode

Value	Description
0	At least one detector flag is necessary to trigger the correction.
1	All detector flags are required to trigger the correction.

Bit 14 – TA_ENABLE Average Threshold Enable

Value	Description
0	Average detector is disabled.
1	Average detector is enabled.

Bit 13 – TC_ENABLE Closest Pixels Threshold Enable

Value	Description
0	Closest Pixels detector is disabled.
1	Closest Pixels detector is enabled.

Bit 12 – TM_ENABLE Median Threshold Enable

Value	Description
0	Median detector is disabled.
1	Median detector is enabled.

Bit 4 – EITPOL Edge Interpolation

Value	Description
0	No edge interpolation is performed.
1	Edge interpolation is performed.

Bits 1:0 – BAYCFG[1:0] Color Filter Array Pattern

Value	Name	Description
0	GRGR	Starting row configuration is G R G R (red row)
1	RGRG	Starting row configuration is R G R G (red row)
2	GBGB	Starting row configuration is G B G B (blue row)
3	BGBG	Starting row configuration is B G B G (blue row)

43.7.17 ISC Defective Pixel Correction Median Threshold Register

Name: ISC_DPC_THRESHM
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					THRESHM[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	THRESHM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – THRESHM[11:0] Median Threshold

43.7.18 ISC Defective Pixel Correction Closest Threshold Register

Name: ISC_DPC_THRESHC
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					THRESHC[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	THRESHC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – THRESHC[11:0] Closest Pixel Threshold

43.7.19 ISC Defective Pixel Correction Average Threshold Register

Name: ISC_DPC_THRESHA
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					THRESHA[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	THRESHA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – THRESHA[11:0] Average Threshold

43.7.20 ISC Defective Pixel Correction Status Register

Name: ISC_DPC_SR
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	COUNTER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – COUNTER[23:0] Defective Pixel Counter (cleared on read)
This field shows the number of active pixel substitutions in the previous frame. It is updated on a frame-by-frame basis.

43.7.21 ISC White Balance Control Register

Name: ISC_WB_CTRL
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE White Balance Enable

Value	Description
0	The white balance is disabled.
1	The white balance is enabled.

43.7.22 ISC White Balance Configuration Register

Name: ISC_WB_CFG
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							BAYCFG[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – BAYCFG[1:0] White Balance Bayer Configuration (Pixel Color Pattern)

Value	Name	Description
0	GRGR	Starting Row configuration is G R G R (Red Row).
1	RGRG	Starting Row configuration is R G R G (Red Row).
2	GBGB	Starting Row configuration is G B G B (Blue Row).
3	BGBG	Starting Row configuration is B G B G (Blue Row).

43.7.23 ISC White Balance Offset for R, GR Register

Name: ISC_WB_O_RGR
Offset: 0x60
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				GROFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GROFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
				ROFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ROFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – GROFST[12:0] Offset Green Component for Red Row (signed 13 bits 1:12:0)

Bits 12:0 – ROFST[12:0] Offset Red Component (signed 13 bits 1:12:0)

43.7.24 ISC White Balance Offset for B and GB Register

Name: ISC_WB_O_BGB
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				GBOFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	GBOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
				BOFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	BOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – GBOFST[12:0] Offset Green Component for Blue Row (signed 13 bits, 1:12:0)

Bits 12:0 – BOFST[12:0] Offset Blue Component (signed 13 bits, 1:12:0)

43.7.25 ISC White Balance Gain for R, GR Register

Name: ISC_WB_G_RGR
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				GRGAIN[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				RGAIN[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – GRGAIN[12:0] Green Component (Red row) Gain (unsigned 13 bits, 0:4:9)

Bits 12:0 – RGAIN[12:0] Red Component Gain (unsigned 13 bits, 0:4:9)

43.7.26 ISC White Balance Gain for B, GB Register

Name: ISC_WB_G_BGB
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				GBGAIN[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				BGAIN[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – GBGAIN[12:0] Green Component (Blue row) Gain (unsigned 13 bits, 0:4:9)

Bits 12:0 – BGAIN[12:0] Blue Component Gain (unsigned 13 bits, 0:4:9)

43.7.27 ISC Color Filter Array Control Register

Name: ISC_CFA_CTRL
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Color Filter Array Interpolation Enable

Value	Description
0	Color Filter Array Interpolation is disabled.
1	Color Filter Array Interpolation is enabled.

43.7.28 ISC Color Filter Array Configuration Register

Name: ISC_CFA_CFG
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					EAL[1:0]			
Access					R/W	R/W		
Reset					0	0		

Bit	7	6	5	4	3	2	1	0
				EITPOL			BAYCFG[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 11:10 – EAL[1:0] Green Channel Edge Adaptive Level

Value	Name	Description
0	GLINEAR	Green plane is linearly interpolated.
1	GMEAN	Green plane is the mean value between the linearly interpolated plane and adaptive method plane.
2	GADAPTIVE	Green plane is interpolated with edge adaptive method.

Bit 4 – EITPOL Edge Interpolation

Value	Description
0	Edges are not interpolated.
1	Edge interpolation is performed.

Bits 1:0 – BAYCFG[1:0] Color Filter Array Pattern

Value	Name	Description
0	GRGR	Starting row configuration is G R G R (red row).
1	RGRG	Starting row configuration is R G R G (red row).
2	GBGB	Starting row configuration is G B G B (blue row).
3	BGBG	Starting row configuration is B G B G (blue row).

43.7.29 ISC Color Correction Control Register

Name: ISC_CC_CTRL
Offset: 0x78
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Color Correction Enable

Value	Description
0	Color correction is disabled.
1	Color correction is enabled.

43.7.30 ISC Color Correction RR RG Register

Name: ISC_CC_RR_RG
Offset: 0x7C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					RRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – RGGAIN[11:0] Green Gain for Red Component (signed 12 bits, 1:3:8)

Bits 11:0 – RRGAIN[11:0] Red Gain for Red Component (signed 12 bits, 1:3:8)

43.7.31 ISC Color Correction RB OR Register

Name: ISC_CC_RB_OR
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				ROFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ROFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					RBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – ROFST[12:0] Red Component Offset (signed 13 bits, 1:12:0)

Bits 11:0 – RBGAIN[11:0] Blue Gain for Red Component (signed 12 bits, 1:3:8)

43.7.32 ISC Color Correction GR GG Register

Name: ISC_CC_GR_GG
Offset: 0x84
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					GGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	GGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
					GRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	GRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – GGGAIN[11:0] Green Gain for Green Component (signed 12 bits, 1:3:8)

Bits 11:0 – GRGAIN[11:0] Red Gain for Green Component (signed 12 bits, 1:3:8)

43.7.33 ISC Color Correction GB OG Register

Name: ISC_CC_GB_OG
Offset: 0x88
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				GOFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					GBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – GOFST[12:0] Green Component Offset (signed 13 bits, 1:12:0)

Bits 11:0 – GBGAIN[11:0] Blue Gain for Green Component (signed 12 bits, 1:3:8)

43.7.34 ISC Color Correction BR BG Register

Name: ISC_CC_BR_BG
Offset: 0x8C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					BGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					BRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – BGGAIN[11:0] Green Gain for Blue Component (signed 12 bits, 1:3:8)

Bits 11:0 – BRGAIN[11:0] Red Gain for Blue Component (signed 12 bits, 1:3:8)

43.7.35 ISC Color Correction BB OB Register

Name: ISC_CC_BB_OB
Offset: 0x90
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				BOFST[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					BBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – BOFST[12:0] Blue Component Offset (signed 13 bits, 1:12:0)

Bits 11:0 – BBGAIN[11:0] Blue Gain for Blue Component (signed 12 bits, 1:3:8)

43.7.36 ISC Gamma Correction Control Register

Name: ISC_GAM_CTRL
Offset: 0x94
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BIPART	RENABLE	GENABLE	BENABLE	ENABLE
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – BIPART Bipartite Table Configuration

Value	Description
0	Bipartite table is disabled. There are 64 points of interpolation from 0 to 4095.
1	Bipartite table is enabled. There are 32 points of interpolation (spacing is 8) from 0 to 255, then there are 30 points of interpolation from 256 to 4095.

Bit 3 – RENABLE Gamma Correction Enable for R Channel

Value	Description
0	12-bit to 10-bit compression is performed skipping two bits.
1	Piecewise interpolation is used to perform 12-bit to 10-bit compression for the red channel.

Bit 2 – GENABLE Gamma Correction Enable for G Channel

Value	Description
0	12-bit to 10-bit compression is performed skipping two bits.
1	Piecewise interpolation is used to perform 12-bit to 10-bit compression for the green channel.

Bit 1 – BENABLE Gamma Correction Enable for B Channel

Value	Description
0	12-bit to 10-bit compression is performed skipping two bits.
1	Piecewise interpolation is used to perform 12-bit to 10-bit compression for the blue channel.

Bit 0 – ENABLE Gamma Correction Enable

Value	Description
0	Gamma correction is disabled.
1	Gamma correction is enabled.

43.7.37 ISC Gamma Correction Blue Entry Register x [x=0..63]

Name: ISC_GAM_BENTRYx
Offset: 0x98 + x*0x04 [x=0..63]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							BCONSTANT[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	BCONSTANT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							BSLOPE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	BSLOPE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – BCONSTANT[9:0] Blue Color Constant for Piecewise Interpolation (unsigned 10 bits 0:10:0)

Bits 9:0 – BSLOPE[9:0] Blue Color Slope for Piecewise Interpolation (signed 10 bits 1:3:6)

43.7.38 ISC Gamma Correction Green Entry Register x [x=0..63]

Name: ISC_GAM_GENTRYx
Offset: 0x0198 + x*0x04 [x=0..63]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							GCONSTANT[9:8]	
Access							R/W	R/W
Reset							0	0
Bit	23	22	21	20	19	18	17	16
	GCONSTANT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
							GSLOPE[9:8]	
Access							R/W	R/W
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	GSLOPE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – GCONSTANT[9:0] Green Color Constant for Piecewise Interpolation (unsigned 10 bits 0:10:0)

Bits 9:0 – GSLOPE[9:0] Green Color Slope for Piecewise Interpolation (signed 10 bits 1:3:6)

43.7.39 ISC Gamma Correction Red Entry Register x [x=0..63]

Name: ISC_GAM_RENTRYx
Offset: 0x0298 + x*0x04 [x=0..63]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							RCONSTANT[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	RCONSTANT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							RSLOPE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RSLOPE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – RCONSTANT[9:0] Red Color Constant for Piecewise Interpolation (unsigned 10 bits 0:10:0)

Bits 9:0 – RSLOPE[9:0] Red Color Slope for Piecewise Interpolation (signed 10 bits 1:3:6)

43.7.40 ISC VHXS Control Register

Name: ISC_VHXS_CTRL
Offset: 0x398
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							HXSEN	VXSEN
Access							R/W	R/W
Reset							0	0

Bit 1 – HXSEN Horizontal Scaler Enable

Value	Description
0	Horizontal scaler is disabled.
1	Horizontal scaler is enabled.

Bit 0 – VXSEN Vertical Scaler Enable

Value	Description
0	Vertical scaler is disabled.
1	Vertical scaler is enabled.

43.7.41 ISC VHXS Source Size Register

Name: ISC_VHXS_SS
Offset: 0x39C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					YS[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	YS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					XS[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	XS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – YS[11:0] Source Image Vertical Size
The horizontal size of the source image is (YS+1) pixels.

Bits 11:0 – XS[11:0] Source Image Horizontal Size
The horizontal size of the source image is (XS+1) pixels.

43.7.42 ISC VHXS Destination Size Register

Name: ISC_VHXS_DS
Offset: 0x3A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					YD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	YD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					XD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – YD[11:0] Destination Image Vertical Size
The vertical size of the destination image is (YD+1) pixels.

Bits 11:0 – XD[11:0] Destination Image Horizontal Size
The horizontal size of the destination image is (XD+1) pixels.

43.7.43 ISC VXS Scaling Factor Register

Name: ISC_VXS_FACT
Offset: 0x3A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	VFACT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	VFACT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VFACT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – VFACT[23:0] Vertical Scaling Factor
The vertical scaling factor format is 0.4.20.

43.7.44 ISC HXS Scaling Factor Register

Name: ISC_HXS_FACT
Offset: 0x3A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HFACT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HFACT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HFACT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – HFACT[23:0] Horizontal Scaling Factor
The horizontal scaling factor format is 0.4.20.

43.7.45 ISC VXS Configuration Register

Name: ISC_VXS_CFG
Offset: 0x3AC
Reset: 0x80000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	FLMAX[3:0]				FLMIN[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					OFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TAP2			FILTCFG[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 31:28 – FLMAX[3:0] Flush Latency Maximum
Defines the maximum number of valid cycles between two lines when the pipeline generates lines.

Bits 27:24 – FLMIN[3:0] Flush Latency Minimum
Defines the minimum number of valid cycles between two lines.

Bits 11:8 – OFFSET[3:0] Resampling Default Phase
Defines the phase initialization of the filter.

Bit 4 – TAP2 Bilinear Interpolation

Value	Description
0	Custom tap values are used (see ISC_VXS_TAP10PHI).
1	Bilinear interpolation is used.

Bits 1:0 – FILTCFG[1:0] Vertical Filter Initial Configuration
Defines how the resampling filter will be initialized. Use value 1 for RGB interpolated pixel stream.

43.7.46 ISC HXS Configuration Register

Name: ISC_HXS_CFG
Offset: 0x3B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					FL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					OFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TAP2			FILTCFG[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 27:24 – FL[3:0] Flush Latency
Defines the minimum number of valid cycles between two lines.

Bits 11:8 – OFFSET[3:0] Resampling Default Phase
Defines the phase initialization of the filter.

Bit 4 – TAP2 Bilinear Interpolation

Value	Description
0	Custom tap values are used (see ISC_HXS_TAP10PHI).
1	Bilinear interpolation is used.

Bits 1:0 – FILTCFG[1:0] Horizontal Filter Initial Configuration
Defines how the resampling filter will be initialized. Use value 1 for RGB interpolated pixel stream.

43.7.47 ISC VXS TAP10 Phase x Register

Name: ISC_VXS_TAP10PHIx
Offset: 0x03B4 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				TAP1[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TAP0[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAP0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – TAP1[12:0] Vertical Filter Tap 1 Coefficient
Filter coefficient for TAP 1, the format is 1.2.10.

Bits 12:0 – TAP0[12:0] Vertical Filter Tap 0 Coefficient
Filter coefficient for TAP 0, the format is 1.2.10.

43.7.48 ISC VXS TAP32 Phase x Register

Name: ISC_VXS_TAP32PHIx
Offset: 0x03B8 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				TAP3[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TAP3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
				TAP2[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TAP2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – TAP3[12:0] Vertical Filter Tap 3 Coefficient
Filter coefficient for TAP 3, the format is 1.2.10.

Bits 12:0 – TAP2[12:0] Vertical Filter Tap 2 Coefficient
Filter coefficient for TAP 2, the format is 1.2.10.

43.7.49 ISC HXS TAP10 Phase x Register

Name: ISC_HXS_TAP10PHIx
Offset: 0x0434 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				TAP1[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TAP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
				TAP0[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TAP0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – TAP1[12:0] Vertical Filter Tap 1 Coefficient
Filter coefficient for TAP 1, the format is 1.2.10.

Bits 12:0 – TAP0[12:0] Vertical Filter Tap 0 Coefficient
Filter coefficient for TAP 0, the format is 1.2.10.

43.7.50 ISC HXS TAP32 Phase x Register

Name: ISC_HXS_TAP32PHIx
Offset: 0x0438 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				TAP3[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	TAP3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
				TAP2[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TAP2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:16 – TAP3[12:0] Vertical Filter Tap 3 Coefficient
Filter coefficient for TAP 3, the format is 1.2.10.

Bits 12:0 – TAP2[12:0] Vertical Filter Tap 2 Coefficient
Filter coefficient for TAP 2, the format is 1.2.10.

43.7.51 ISC Color Space Conversion Control Register

Name: ISC_CSC_CTRL
Offset: 0x4B4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE RGB to YCbCr Color Space Conversion Enable

Value	Description
0	Color space conversion is disabled.
1	Color space conversion is enabled.

43.7.52 ISC Color Space Conversion YR YG Register

Name: ISC_CSC_YR_YG
Offset: 0x4B8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					YGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	YGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					YRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	YRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – YGGAIN[11:0] Green Gain for Luminance (signed 12 bits 1:3:8)

Bits 11:0 – YRGAIN[11:0] Reg Gain for Luminance (signed 12 bits 1:3:8)

43.7.53 ISC Color Space Conversion YB OY Register

Name: ISC_CSC_YB_OY
Offset: 0x4BC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YOFST[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	23	22	21	20	19	18	17	16
	YOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					YBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	YBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YOFST[10:0] Luminance Offset (11 bits signed 1:10:0)

Bits 11:0 – YBGAIN[11:0] Blue Gain for Luminance Component (12 bits signed 1:3:8)

43.7.54 ISC Color Space Conversion CBR CBG Register

Name: ISC_CSC_CBR_CBG
Offset: 0x4C0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					CBGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CBGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CBRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CBRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – CBGGAIN[11:0] Green Gain for Blue Chrominance (signed 12 bits 1:3:8)

Bits 11:0 – CBRGAIN[11:0] Red Gain for Blue Chrominance (signed 12 bits, 1:3:8)

43.7.55 ISC Color Space Conversion CBB OCB Register

Name: ISC_CSC_CBB_OCB
Offset: 0x4C4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						CBOFST[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	CBOFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CBBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CBBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – CBOFST[10:0] Blue Chrominance Offset (signed 11 bits 1:10:0)

Bits 11:0 – CBBGAIN[11:0] Blue Gain for Blue Chrominance (signed 12 bits 1:3:8)

43.7.56 ISC Color Space Conversion CRR CRG Register

Name: ISC_CSC_CRR_CRG
Offset: 0x4C8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					CRGGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRGGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CRRGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – CRGGAIN[11:0] Green Gain for Red Chrominance (signed 12 bits 1:3:8)

Bits 11:0 – CRRGAIN[11:0] Red Gain for Red Chrominance (signed 12 bits 1:3:8)

43.7.57 ISC Color Space Conversion CRB OCR Register

Name: ISC_CSC_CRB_OCR
Offset: 0x4CC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						CROFST[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	23	22	21	20	19	18	17	16
	CROFST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					CRBGAIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRBGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – CROFST[10:0] Red Chrominance Offset (signed 11 bits 1:10:0)

Bits 11:0 – CRBGAIN[11:0] Blue Gain for Red Chrominance (signed 12 bits 1:3:8)

43.7.58 ISC Control Register

Name: ISC_CBHS_CTRL
Offset: 0x4D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Contrast, Brightness, Hue and Saturation Control Enable

Value	Description
0	CBHS control is disabled.
1	CBHS control is enabled.

43.7.59 ISC CBHS Configuration Register

Name: ISC_CBHS_CFG
Offset: 0x4D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CCIRMODE[1:0]		CCIR
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – CCIRMODE[1:0] CCIR656 Byte Ordering

Value	Name	Description
0	CBY	Byte ordering Cb0, Y0, Cr0, Y1
1	CRY	Byte ordering Cr0, Y0, Cb0, Y1
2	YCB	Byte ordering Y0, Cb0, Y1, Cr0
3	YCR	Byte ordering Y0, Cr0, Y1, Cb0

Bit 0 – CCIR CCIR656 Stream Enable

Value	Description
0	Raw mode.
1	CCIR656 stream.

43.7.60 ISC CBHS Brightness Register

Name: ISC_CBHS_BRIGHT
Offset: 0x4D8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						BRIGHT[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	BRIGHT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 10:0 – BRIGHT[10:0] Image Brightness Control (signed 11 bits 1:10:0)
Brightness value is added or subtracted from the luminance Y data.

43.7.61 ISC CBHS Contrast Register

Name: ISC_CBHS_CONTRAST
Offset: 0x4DC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CONTRAST[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONTRAST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – CONTRAST[11:0] Contrast (unsigned 12 bits 0:4:8)
Adjusts the image contrast by multiplying with YCbCr data.

43.7.62 ISC Hue Register

Name: ISC_CBHS_HUE
Offset: 0x4E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								HUE[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	HUE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – HUE[8:0] Image Hue Value (unsigned 9 bits 0:9:0)
Programs the hue with angle in degree the range is [0:359] The hue control is implemented by mixing Cb and Cr data.

43.7.63 ISC CBHS Saturation Register

Name: ISC_CBHS_SAT
Offset: 0x4E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					SATURATION[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SATURATION[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – SATURATION[11:0] Image Saturation Value (unsigned 12 bits 0:8:4)
Saturation is adjusted by multiplying both Cb and Cr by a constant.

43.7.64 ISC Subsampling 4:4:4 to 4:2:2 Control Register

Name: ISC_SUB422_CTRL
Offset: 0x4E8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE 4:4:4 to 4:2:2 Chrominance Horizontal Subsampling Filter Enable

Value	Description
0	Subsampler is disabled.
1	Subsampler is enabled.

43.7.65 ISC Subsampling 4:4:4 to 4:2:2 Configuration Register

Name: ISC_SUB422_CFG
Offset: 0x4EC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			FILTER[1:0]			CCIRMODE[1:0]		CCIR
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bits 5:4 – FILTER[1:0] Low Pass Filter Selection

Value	Name	Description
0	FILT0CO	Cosited, {1}
1	FILT1CE	Centered {1, 1}
2	FILT2CO	Cosited {1,2,1}
3	FILT3CE	Centered {1, 3, 3, 1}

Bits 2:1 – CCIRMODE[1:0] CCIR656 Byte Ordering

Value	Name	Description
0	CBY	Byte ordering Cb0, Y0, Cr0, Y1
1	CRY	Byte ordering Cr0, Y0, Cb0, Y1
2	YCB	Byte ordering Y0, Cb0, Y1, Cr0
3	YCR	Byte ordering Y0, Cr0, Y1, Cb0

Bit 0 – CCIR CCIR656 Input Stream

Value	Description
0	Raw mode.
1	CCIR mode.

43.7.66 ISC Subsampling 4:2:2 to 4:2:0 Control Register

Name: ISC_SUB420_CTRL
Offset: 0x4F0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MIPI420	FILTER				ENABLE
Access			R/W	R/W				R/W
Reset			0	0				0

Bit 5 – MIPI420 MIPI YUV 420 8-bpp or 10-bpp Even Odd Splitter

Value	Description
0	Normal mode.
1	When the MIPI interface is selected and the source format is YUV 420 RMS (recommended memory storage), the sub420 submodule routes data lanes depending on the parity of the line received (odd or even).

Bit 4 – FILTER Interlaced or Progressive Chrominance Filter

Value	Description
0	Progressive filter {0.5, 0.5}.
1	Field-dependent filter, top field filter is {0.75, 0.25}, bottom field filter is {0.25, 0.75}.

Bit 0 – ENABLE 4:2:2 to 4:2:0 Vertical Subsampling Filter Enable (Center Aligned)

Value	Description
0	Subsampler disabled.
1	Subsampler enabled.

43.7.67 ISC Rounding, Limiting and Packing Configuration Register

Name: ISC_RLP_CFG
Offset: 0x4F4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	ALPHA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	YMODE[1:0]		LSH	REP	MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – ALPHA[7:0] Alpha Value for Alpha-enabled RGB Mode
This field is relevant for ARGB444, ARGB555 and ARGB32 pixel formats.

Bits 7:6 – YMODE[1:0] YCbCr Memory Mapping Configuration Mode
YMODE is only available for YCYC and YCYC_Limited modes.

Value	Name	Description
0	RLP_YCBYCR	Byte 0 is Cr, Byte 1 is Y(n), Byte 2 is Cb, Byte 3 is Y(n+1)
1	RLP_YCRYCB	Byte 0 is Cb, Byte 1 is Y(n), Byte 2 is Cb, Byte 3 is Y(n+1)
2	RLP_CBYCRY	Byte 0 is Y(n), Byte 1 is Cr, Byte 2 is Y(n+1), Byte 3 is Cb
3	RLP_CRYCBY	Byte 0 is Y (n), Byte 1 is Cb, Byte 2 is Y(n+1), Byte 3 is Cr

Bit 5 – LSH Logical Left Shift for Pixel to 16-bit Container Mapping

Value	Description
0	Logical left shift is disabled.
1	Pixel value is left-justified in a 16-bit container.

Bit 4 – REP Pixel Expansion with Replication Logic

Value	Description
0	Replication is disabled.
1	Replication is enabled.

Bits 3:0 – MODE[3:0] Rounding, Limiting and Packing Mode

Value	Name	Description
0	DAT8	8-bit data
1	DAT9	9-bit data
2	DAT10	10-bit data
3	DAT11	11-bit data
4	DAT12	12-bit data

Value	Name	Description
5	DATY8	8-bit luminance only
6	DATY10	10-bit luminance only
7	ARGB444	12-bit RGB+4-bit Alpha (MSB)
8	ARGB555	15-bit RGB+1-bit Alpha (MSB)
9	RGB565	16-bit RGB
10	ARGB32	24-bits RGB mode+8-bit Alpha
11	YYCC	YCbCr mode (full range, [0–255])
12	YYCC_LIMITED	YCbCr mode (limited range)
13	YCYC	Y(n+1)CbY(n)Cr 422 interleaved full range per component 8-bit [0–255]
14	YCYC_LIMITED	Y(n+1)CbY(n)Cr 422 interleaved limited range per component 8-bit
15	BYPASS	32-bit input is sampled and written to the rlp output port. Select this mode for MIPI RMS mode.

43.7.68 ISC Histogram Control Register

Name: ISC_HIS_CTRL
Offset: 0x4F8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Histogram Sub Module Enable

Value	Description
0	Histogram disabled.
1	Histogram enabled.

43.7.69 ISC Histogram Configuration Register

Name: ISC_HIS_CFG
Offset: 0x4FC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								RAR
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BAYSEL[1:0]			MODE[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit 8 – RAR Histogram Reset After Read

Value	Description
0	Reset after read mode is disabled.
1	Reset after read mode is enabled.

Bits 5:4 – BAYSEL[1:0] Bayer Color Component Selection

Value	Name	Description
0	GRGR	Starting row configuration is G R G R (red row)
1	RGRG	Starting row configuration is R G R G (red row)
2	GBGB	Starting row configuration is G B G B (blue row)
3	BGBG	Starting row configuration is B G B G (blue row)

Bits 2:0 – MODE[2:0] Histogram Operating Mode

Value	Name	Description
0	GR	Gr sampling
1	R	R sampling
2	GB	Gb sampling
3	B	B sampling
4	Y	Luminance-only mode
5	RAW	Raw sampling
6	YCCIR656	Luminance only with CCIR656 10-bit or 8-bit mode

43.7.70 ISC DMA Configuration Register

Name: ISC_DCFG
Offset: 0x51C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	AWQOS[3:0]				ARQOS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						CMBSIZE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		YMBSIZE[2:0]				IMODE[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 23:20 – AWQOS[3:0] Write QoS Value
If AWQOS is set to 0, the QoS value depends on the output FIFO level (dynamic configuration). Otherwise the value is defined in the register.

Bits 19:16 – ARQOS[3:0] Read QoS Value
Returns the QoS bus value when a descriptor is retrieved from the memory.

Bits 10:8 – CMBSIZE[2:0] DMA Memory Burst Size C channel

Value	Name	Description
0	SINGLE	DMA single access
1	BEATS4	4-beat burst access
2	BEATS8	8-beat burst access
3	BEATS16	16-beat burst access
4	BEATS32	32-beat burst access

Bits 6:4 – YMBSIZE[2:0] DMA Memory Burst Size Y channel

Value	Name	Description
0	SINGLE	DMA single access
1	BEATS4	4-beat burst access
2	BEATS8	8-beat burst access
3	BEATS16	16-beat burst access
4	BEATS32	32-beat burst access

Bits 2:0 – IMODE[2:0] DMA Input Mode Selection

Value	Name	Description
0	PACKED8	8 bits, single channel packed
1	PACKED16	16 bits, single channel packed
2	PACKED32	32 bits, single channel packed

Value	Name	Description
3	YC422SP	32 bits, dual channel
4	YC422P	32 bits, triple channel
5	YC420SP	32 bits, dual channel
6	YC420P	32 bits, triple channel

43.7.71 ISC DMA Control Register

Name: ISC_DCTRL
Offset: 0x520
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DONE	FIELD	WB	IE		DVIEW[1:0]		DE
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – DONE Descriptor Processing Status
This bit appears only in the descriptor located in memory and can be used only if WB (Write Back) is set.

Value	Description
0	Descriptor not processed yet.
1	Descriptor processed.

Bit 6 – FIELD Value of Captured Frame Field Signal
This bit is only relevant for interlaced content. It appears only in the descriptor located in memory and can be used only if WB (Write Back) is set.

Value	Description
0	Field value is 0.
1	Field value is 1.

Bit 5 – WB Write Back Operation Enable

Value	Description
0	Write Back operation is skipped.
1	Write Back operation is performed.

Bit 4 – IE Interrupt Enable

Value	Description
0	DMA Done interrupt is generated.
1	DMA Done interrupt is not set.

Bits 2:1 – DVIEW[1:0] Descriptor View

Value	Name	Description
0	PACKED	Address {0} Stride {0} are updated
1	SEMIPLANAR	Address {0,1} Stride {0,1} are updated
2	PLANAR	Address {0,1,2} Stride {0,1,2} are updated

Bit 0 – DE Descriptor Enable

Value	Description
0	Descriptor disabled.
1	Descriptor enabled.

43.7.72 ISC DMA Descriptor Address Register

Name: ISC_DNDA
Offset: 0x524
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NDA[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NDA[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NDA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NDA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – NDA[29:0] Next Descriptor Address Register

43.7.73 ISC DMA Address 0 Register

Name: ISC_DAD0
Offset: 0x528
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	AD0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	AD0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	AD0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	AD0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AD0[31:0] Channel 0 Address

43.7.74 ISC DMA Stride 0 Register

Name: ISC_DST0
Offset: 0x52C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ST0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ST0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ST0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ST0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ST0[31:0] Channel 0 Stride

43.7.75 ISC DMA Address 1 Register

Name: ISC_DAD1
Offset: 0x530
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	AD1[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AD1[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AD1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AD1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AD1[31:0] Channel 1 Address

43.7.76 ISC DMA Stride 1 Register

Name: ISC_DST1
Offset: 0x534
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ST1[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ST1[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ST1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ST1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ST1[31:0] Channel 1 Stride

43.7.77 ISC DMA Address 2 Register

Name: ISC_DAD2
Offset: 0x538
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	AD2[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AD2[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AD2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AD2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AD2[31:0] Channel 2 Address

43.7.78 ISC DMA Stride 2 Register

Name: ISC_DST2
Offset: 0x53C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ST2[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ST2[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ST2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ST2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ST2[31:0] Channel 2 Stride

43.7.79 ISC Histogram Entry x [x=0..511]

Name: ISC_HIS_ENTRYx
Offset: 0x055C + x*0x04 [x=0..511]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					COUNT[19:16]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

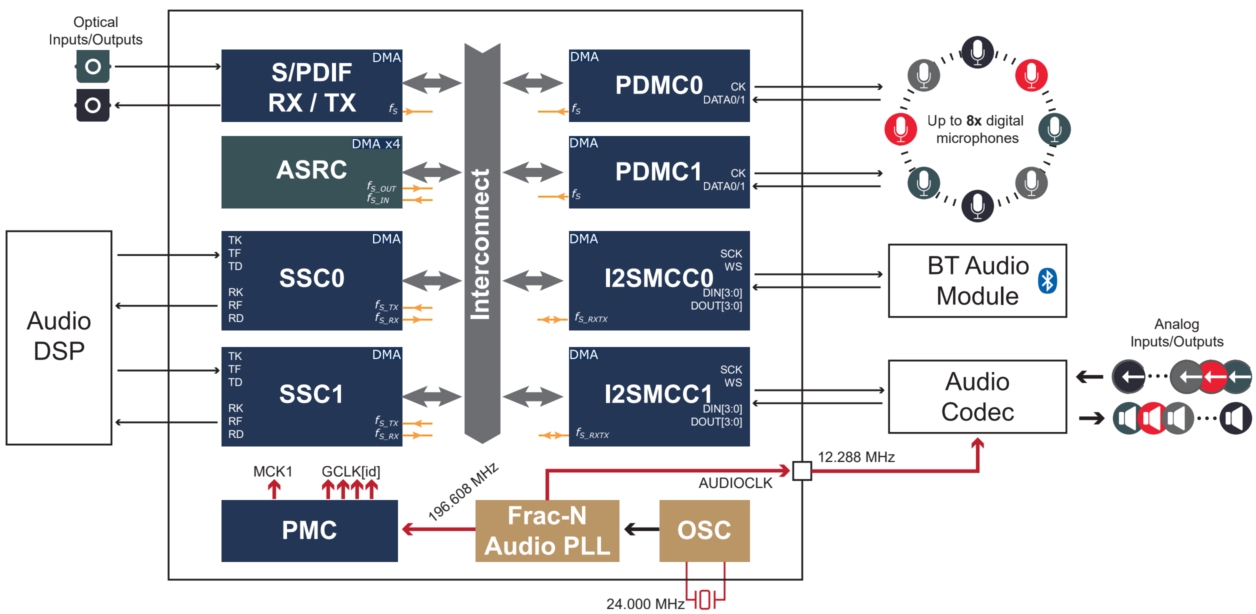
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:0 – COUNT[19:0] Entry Counter

44. AUDIO SUBSYSTEM

44.1 Block Diagram

Figure 44-1. Audio Subsystem Block Diagram Example



44.2 Components

The audio subsystem is made of the following components. Unless otherwise specified, all are designed to handle audio sample rates up to 192 kHz.

- 2x Synchronous Serial Controller (SSC)
  - The maximum bit clock (TK or RK) speed is 25 MHz in Host or Client mode on both the receiver and the transmitter.
  - As an example, in TDM mode, this corresponds to up to 4x 32-bit channels at 192 kHz, or up to 8x 32-bit channels at 96 kHz.
- 2x Inter-IC Sound Multi Channel Controller (I2SMCC)
  - The maximum bit clock (I2SMCCx_SCK) speed is 12.5 MHz in both Host and Client modes.
  - In TDM256 mode, this corresponds to 8x 32-bit channels at 48 kHz.
  - In 4x data lines Stereo mode, up to 8x 32-bit channels at 192 kHz are supported.
- 2x Pulse Density Microphone Controller (PDMC)
  - Support of up to 8x microphones on 4 data lines with 48 kHz sampling rate
- 1x Sony/Philips Digital Interface Receiver (SPDIFRX)
  - Support of sample rate up to 192 kHz with $f_{GCLK} \geq 150$ MHz
- 1x Sony/Philips Digital Interface Transmitter (SPDIFTX)
  - Support of sample rate up to 192 kHz stereo with $f_{GCLK} \geq 24.5476$ MHz
- 1x low-jitter audio PLL with one dedicated AUDIOCLK output pin
- 1x Asynchronous Sample Rate Converter (ASRC) with 4 stereo channels
  - Supporting up to 4 independent conversion rates

44.3 Product Dependencies

44.3.1 Clocks

Audio peripherals are part of the APB Client (APS) matrix, clocked by MCK1. Their peripheral clocks and generic clocks are controlled in the PMC.

To generate audio frequencies with high accuracy, the device embeds a fractional audio PLL that can serve both the internal (GCLK generation) and external needs (MCK generation on external audio components).

The ASRC receives the frame synchronization signals from all audio peripherals so it can adapt its internal DSP filters to the (unknown) rate of the interface it is connected to, on both its input and output sides. Therefore, converting the audio sample rate from one interface to another can be done with no knowledge of these rates.

44.3.2 Interrupts

Refer to the table Peripheral Identifiers.

44.3.3 Reset

Audio peripherals are connected to the processor and peripherals reset line.

44.3.4 I/Os

Audio I/Os are multiplexed on GPIOs with various power supplies. For the applicable I/O type and power supply, refer to the table Pin Description.

44.4 Special Functions in SFR/SFRBU

None.

45. Inter-IC Sound Multi-Channel Controller (I2SMCC)

45.1 Description

The Inter-IC Sound Multi-Channel Controller (I2SMCC) provides an 11-wire, bidirectional, synchronous, digital audio link to external audio devices: I2SMCC_DIN3..0, I2SMCC_DOUT3..0, I2SMCC_WS, I2SMCC_CK, and I2SMCC_MCK pins.

The I2SMCC complies with the Inter-IC Sound (I^2S) bus specification and supports a Time Division Multiplexed (TDM) interface with external multi-channel audio codecs.

The I2SMCC consists of a receiver, a transmitter and a common clock generator that can be enabled separately to provide Host, Client or Controller modes with receiver and/or transmitter active.

DMA Controller channels, separate for the receiver and for the transmitter, allow a continuous high bit rate data transfer without processor intervention to the following:

- Audio CODECs in Host, Client, or Controller mode
- Stereo DAC or ADC through a dedicated I^2S serial interface
- Multi-channel or multiple stereo DACs or ADCs, using the TDM format

The I2SMCC uses a single DMA Controller channel for all audio channels.

The 8- and 16-bit compact stereo formats reduce the required DMA Controller bandwidth by transferring the left and right samples within the same data word.

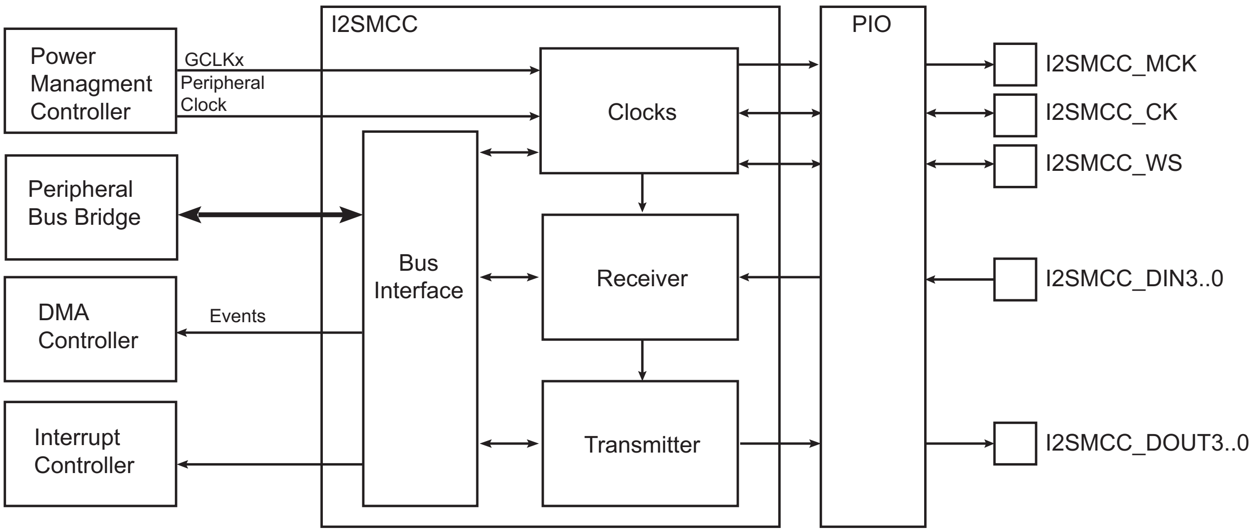
In Host mode, the I2SMCC can produce a 16 f_s to 1024 f_s host clock that provides an over-sampling clock to an external audio codec or digital signal processor (DSP).

45.2 Embedded Characteristics

- Compliant with Inter-IC Sound (I^2S) Bus Specification
- Host, Client, and Controller Modes
  - Client: Data received/transmitted
  - Host: Data received/transmitted and clocks generated
  - Controller: Clocks generated
- FIFO
- Individual Enable and Disable of Receiver, Transmitter and Clocks
- Configurable Clock Generator Common to Receiver and Transmitter
  - Suitable for a wide range of sample frequencies (f_s), including 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, and 192 kHz
  - 16 f_s to 1024 f_s host clock generated for external oversampling data converters
- Support for Multiple Data Formats
  - 32-, 24-, 20-, 18-, 16-, and 8-bit mono or stereo format
  - 16- and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers
- Support for Multiple Data Frame Formats
  - 2-channel I^2S with word select
  - 1- to 8-channel Time Division Multiplexed (TDM) with frame synchronization
- DMA Controller Interfaces the Receiver and Transmitter to Reduce Processor Overhead
- Smart Holding Registers Management to Avoid Audio Channel Mix After Overrun or Underrun
- Functional Safety Monitors and Reports
  - Internal sequencer integrity check reports
  - Register write protection

45.3 Block Diagram

Figure 45-1. I2SMCC Block Diagram



45.4 I/O Lines Description

Table 45-1. I/O Lines Description

Pin Name	Pin Description	Type
I2SMCC_MCK	Host Clock	Output
I2SMCC_CK	Serial Clock	Input/Output
I2SMCC_WS	I^2S Word Select or TDM Frame Synchronization	Input/Output
I2SMCC_DIN3..0	Serial Data Inputs	Input
I2SMCC_DOUT3..0	Serial Data Outputs	Output

45.5 Product Dependencies

To use the I2SMCC, other parts of the system must be configured correctly, as described below.

45.5.1 I/O Lines

The I2SMCC pins may be multiplexed with I/O Controller lines. The user must first program the PIO Controller to assign the required I2SMCC pins to their peripheral function. If the I2SMCC I/O lines are not used by the application, they can be used for other purposes by the PIO Controller. The user must enable the I2SMCC inputs and outputs that are used.

45.5.2 Power Management

If the processor enters a Sleep mode that disables clocks used by the I2SMCC, the I2SMCC stops functioning and resumes operation after the system wakes up from Sleep mode.

45.5.3 Clocks

The I2SMCC runs from the peripheral clock and the generic clock (GCLK), both generated by the Power Management Controller (PMC). Prior to using the I2SMCC, the user must first program the PMC. The I^2S host and serial clock can be generated either from the peripheral clock or the generic clock.

In a similar way, the I2SMCC must be disabled before removing its clock source to avoid freezing it in an undefined state.

45.5.4 DMA Controller

The I2SMCC interfaces to the DMA Controller. Using the I2SMCC DMA functionality requires the DMA Controller to be programmed first.

45.5.5 Interrupt Sources

The I2SMCC interrupt line is connected to the Interrupt Controller. Using the I2SMCC interrupt requires the Interrupt Controller to be programmed first.

45.6 Functional Description

45.6.1 Initialization

The I2SMCC features a receiver, a transmitter and a clock generator for Host and Controller modes. Receiver and transmitter share the same serial clock and word select.

Before enabling the I2SMCC, the selected configuration must be written to the I2SMCC Mode Register A (I2SMCC_MRA). If I2SMCC_MRA.FORMAT is configured in one of the TDM formats, then the I2SMCC_MRA.NBCHAN and I2SMCC_MRA.TDMFS fields must also be written.

Once the I2SMCC_MRA has been written, the I2SMCC clock generator, receiver, and transmitter can be enabled by writing a ‘1’ to the CKEN, RXEN, and TXEN bits in the Control Register (I2SMCC_CR). The clock generator can be enabled alone in Controller mode to output clocks to the I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins. The clock generator must also be enabled if the receiver or the transmitter is enabled.

The clock generator, receiver, and transmitter can be disabled independently by writing a ‘1’ to I2SMCC_CR.CXDIS, I2SMCC_CR.RXDIS and/or I2SMCC_CR.TXDIS, respectively. Once requested to stop, they stop only when the transmission of the pending frame transmission is completed.

45.6.2 Basic Operation

The following description is only valid when the FIFO is disabled (I2SMCC_MRB.FIFOEN = 0).

When the FIFO is enabled, the flags RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx are not relevant in I2SMCC_ISRA. For details, refer to Holding Registers.

The receiver can be operated by reading the Receiver Holding register (I2SMCC_RHR) or the Receiver Holding Left x registers (I2SMCC_RHLxR) and the Receiver Holding Right x registers (I2SMCC_RHRxR), whenever the Receive Left x Ready (RXLRDYx) bit or the Receive Right Ready (RXRRDYx) bit in the Interrupt Status register A (I2SMCC_ISRA) is set. Successive values read from I2SMCC_RHR correspond to the samples from the first left audio channel to the last left audio channel enabled then from the first right audio channel to the last right audio channel enabled, or from channels 0 to I2SMCC_MRA.NBCHAN in TDM mode for the successive frames.

The transmitter can be operated by writing to the Transmitter Holding register (I2SMCC_THR) or the Transmitter Holding Left x registers (I2SMCC_THLxR) and the Transmitter Holding Right x registers (I2SMCC_THRxR), whenever the Transmit Left x Ready (TXLRDYx) bit or the Transmit Right x Ready (TXRRDYx) bit in the I2SMCC_ISRA is set. Successive values written to I2SMCC_THR correspond to the samples from the first left audio channel to the last left audio channel enabled, then from the first right audio channel to the last right audio channel enabled, or from channels 0 to I2SMCC_MRA.NBCHAN in TDM mode for the successive frames.

The RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx bits can be polled by reading the I2SMCC_ISRA.

The I2SMCC processor load can be reduced by enabling interrupt-driven operation. The RXLRDYx, RXRRDYx, TXLRDYx and/or TXRRDYx interrupt requests can be enabled by writing a ‘1’ to the corresponding bit in the Interrupt Enable Register A (I2SMCC_IERA). The interrupt service routine associated to the I2SMCC interrupt request is executed when at least one of the RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx status bits is set.

45.6.3 Host, Controller and Client Modes

In Host and Controller modes, the I2SMCC provides the host clock, the serial clock and the word select. I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins are outputs.

In Controller mode, the I2SMCC receiver and transmitter are disabled. Only the clocks are enabled and used by an external receiver and/or transmitter.

In Client mode, the I2SMCC receives the serial clock and the word select from an external host. I2SMCC_CK and I2SMCC_WS pins are inputs.

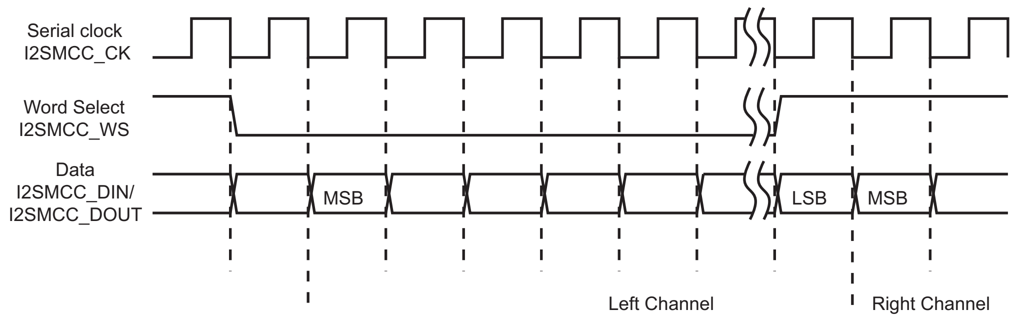
The mode is selected by writing I2SMCC_MRA.MODE. Since the MODE field changes the direction of the I2SMCC_WS and I2SSCK pins, the I2SMCC_MRA must only be written when the I2SMCC is stopped in order to avoid unwanted glitches on the I2SMCC_WS and I2SMCC_CK pins.

Note: When the Client mode is configured and TDM format is selected, the high level duration of the WS input signal can be one-slot (1 sample duration), half-slot or one-bit. The duration must be configured in I2SMCC_MRA.TDMFS (See I2SMCC_MRA).

45.6.4 I²S Reception and Transmission Sequence

As specified in the I²S protocol, data bits are left-justified in the word select time slot, with the MSB transmitted first, starting one clock period after the transition on the word select line.

Figure 45-2. I²S Reception and Transmission Sequence



Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The word select line indicates the channel in transmission, a low level for the left channel and a high level for the right channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing I2SMCC_MRA.DATALENGTH.

If the time slot allows for more data bits than written in I2SMCC_MRA.DATALENGTH, zeroes are appended to the transmitted data word or extra received bits are discarded.

The slot length is defined in the following table.

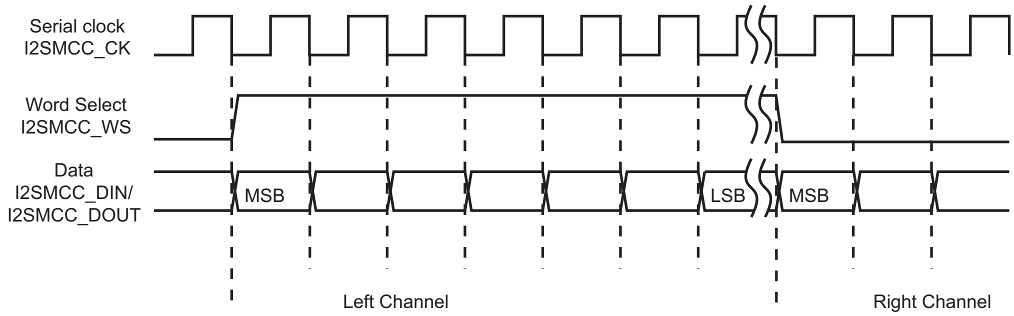
Table 45-2. Slot Length (I²S format)

I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SMCC_MRA.IWS = 0 24 if I2SMCC_MRA.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	
6	8 bits	8
7	8 bits compact stereo	

45.6.5 Left-Justified Reception and Transmission Sequence

As specified in the I²S protocol, data bits are left-justified in the word select time slot, with the MSB transmitted first, starting at the same clock period as the transition on the word select line.

Figure 45-3. Left-Justified Reception and Transmission Sequence



Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The word select line indicates the channel in transmission, a low level for the right channel and a high level for the left channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing I2SMCC_MRA.DATALENGTH.

If the time slot allows for more data bits than written in I2SMCC_MRA.DATALENGTH, zeroes are appended to the transmitted data word or extra received bits are discarded.

The Slot Length is defined in the following table.

Table 45-3. Slot Length (Left-Justified format)

I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SMCC_MRA.IWS = 0 24 if I2SMCC_MRA.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	
6	8 bits	8
7	8 bits compact stereo	

45.6.6 TDM Reception and Transmission Sequence

In Time Division Multiplexed (TDM) format, one to eight data words are sent or received within each frame. As specified in the I^{2}S protocol, data bits are left-justified in the channel time slot, with the MSB transmitted first, starting one clock period after the transition on the word select line. Each time slot is 32 bits long.

Figure 45-4. TDM Reception and Transmission Sequence

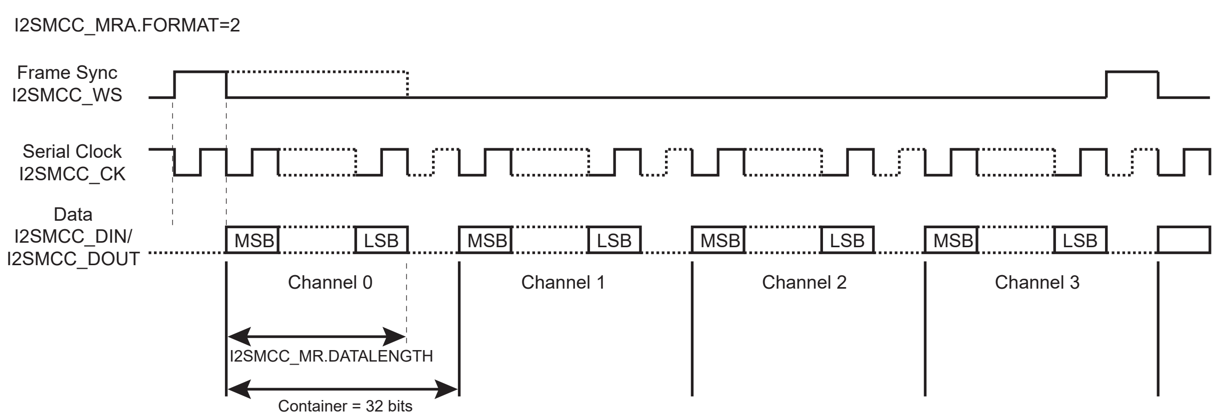
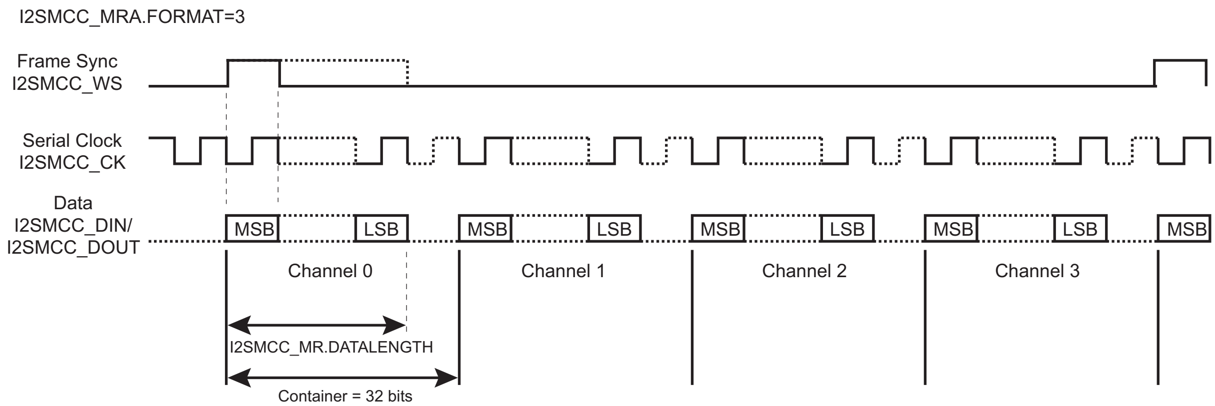


Figure 45-5. TDM Left-Justified Reception and Transmission Sequence



Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The I2SMCC_WS pin provides a frame synchronization signal, starting one I2SMCC_CK period before the MSB of channel 0.

The TDM format is selected when I2SMCC_MRA.FORMAT=2.

The TDM Left-Justified format is selected when I2SMCC_MRA.FORMAT=3.

The Frame Synchronization pulse can be either one I2SMCC_CK period, 16-bit I2SMCC_CK period (half time slot) or one 32-bit time slot. This selection is done by writing the I2SMCC_MRA.TDMFS bit.

The number of channels is selected by writing the I2SMCC_MRA.NBCHAN field.

The Frame Synchronization pulse set to 32-bit time slot with the number of channel set to 1 configuration is not supported.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the I2SMCC_MRA.DATALENGTH field.

If the time slot allows for more data bits than programmed in the I2SMCC_MRA.DATALENGTH field, zeroes are appended to the transmitted data word or extra received bits are discarded.

45.6.7 Serial Clock and Word Select Generation

The generation of clocks in the I2SMCC is described in the following figure.

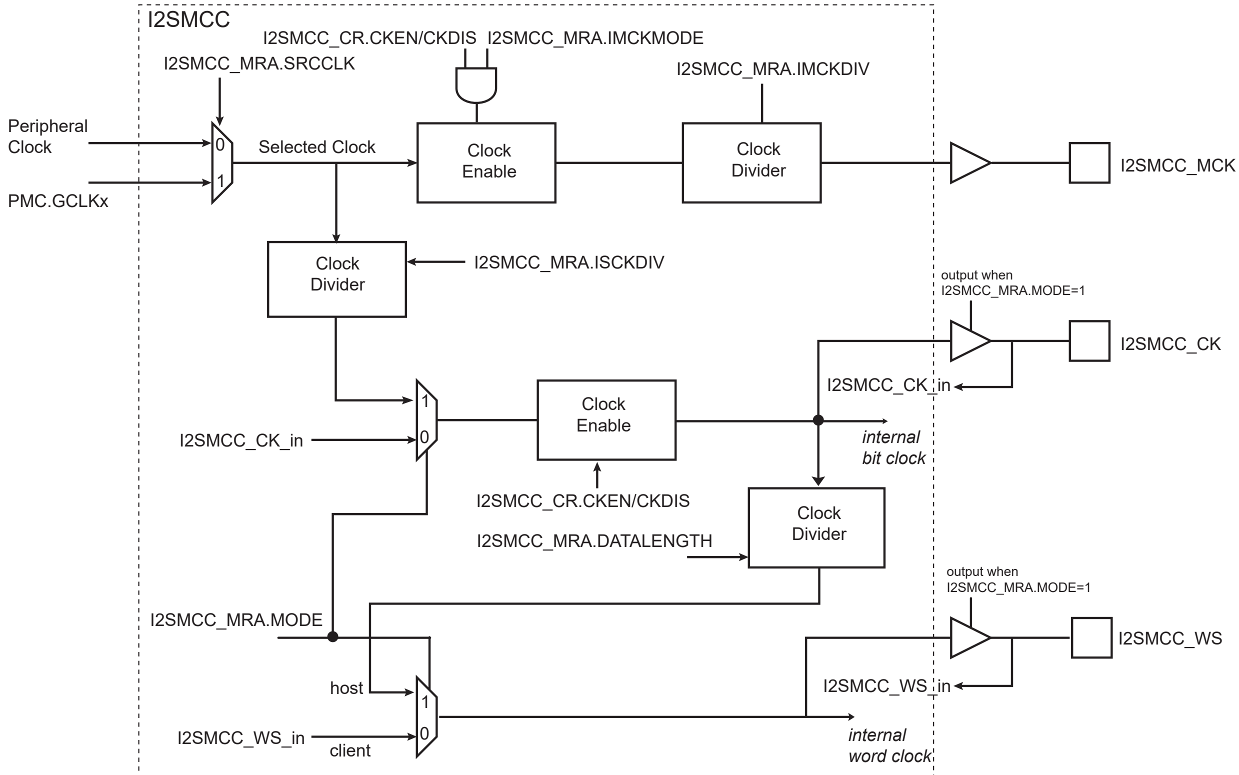
In Client mode, the word select clock is driven by an external host and the serial clock is driven by either an external host or the internal clock circuitry. I2SMCC_CK and I2SMCC_WS pins are inputs.

In Host mode, the user configures the host clock, serial clock, and word select clock through the I2SMCC_MRA. I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins are outputs, MCK and SCK frequencies are configured independently using IMCKDIV and ISCKDIV bits of I2SMCC_MRA, respectively.

If a host clock output is not required, the MCK clock is used as I2SMCC_CK by clearing I2SMCC_MRA.IMCKMODE.

The I2SMCC_WS pin is used as word select in I^{2}S format and as frame synchronization in TDM format, as described in I2S Reception and Transmission Sequence and TDM Reception and Transmission Sequence, respectively.

Figure 45-6. I2SMCC Clock Generation



45.6.8 Mono

When the Transmit Mono bit (TXMONO) in I2SMCC_MRA is set, data written to the left channel is duplicated to the right output channel. In TDM mode, with more than two channels numbered from 0, data written to the even-numbered channels is duplicated to the following odd-numbered channel. When TXMONO is set, TXRRDYx bits of I2SMCC_ISRA are always read as ‘0’ and the behavior of TXRUNFx bits is unchanged.

When the Receive Mono bit (RXMONO) in I2SMCC_MRA is set, data received from the left channel is duplicated to the right input channel. In TDM mode, with more than two channels numbered from 0, data received from the even-numbered channels is duplicated to the following odd-numbered channel. When RXMONO is set, the behavior of RXRRDYx and RXROVFx bits is unchanged.

45.6.9 Wire Configurations

By configuring I^2S or Left-Justified format, three wire configurations are available. The wire configuration is programmable by writing the field I2SMCC_MRA.WIRECFG. The available configurations are defined in the following table.

Table 45-4. Wire Configurations (I^2S and Left-Justified formats)

I2SMCC_MRA.WIRECFG	Configuration	Description
0	1 wire	I2SMCC_DIN0 and I2SMCC_DOUT0 are enabled. Only two I^2S channels are enabled (Left 0 and Right 0).
1	2 wires	I2SMCC_DIN1..0 and I2SMCC_DOUT1..0 are enabled. Only four I^2S channels are enabled (Left 0, Left1, Right 0 and Right 1).
2	4 wires	I2SMCC_DIN3..0 and I2SMCC_DOUT3..0 are enabled. All I^2S channels are enabled.

With the TDM or TDM Left-Justified format selected, four wire configurations are available. The wire configuration is selected via the WIRECFG bit. The available configurations are defined in the following table.

Table 45-5. Wire Configurations (TDM and TDM Left-Justified formats)

I2SMCC_MRA.WIRECFG	Configuration	Description
0	Wire 0	TDM communications take place on I2SMCC_DIN0 and I2SMCC_DOUT0 wires.
1	Wire 1	TDM communications take place on I2SMCC_DIN1 and I2SMCC_DOUT1 wires.
2	Wire 2	TDM communications take place on I2SMCC_DIN2 and I2SMCC_DOUT2 wires.
3	Wire 3	TDM communications take place on I2SMCC_DIN3 and I2SMCC_DOUT3 wires.

45.6.10 Holding Registers

The I2SMCC user interface includes two common holding registers—the Receive Holding Register (I2SMCC_RHR) and the Transmit Holding Register (I2SMCC_THR)— and dedicated holding registers per IO—the Receive Holding Left x Registers (I2SMCC_RHLxR), the Receive Holding Right x Registers (I2SMCC_RHRxR), the Transmit Holding Left x Registers (I2SMCC_THLxR) and the Transmit Holding Right x Registers (I2SMCC_THRxR). The common registers are used to access audio samples for all audio channels; the dedicated registers are used to access audio samples of one specific channel.

Access through common registers and dedicated registers must not be mixed. Only common registers or dedicated registers must be used.

The I2SMCC includes a Transmit (TX) FIFO and a Receive (RX) FIFO. Both FIFOs are only available through the common registers and must be enabled by setting the FIFOEN bit in the Mode register B (I2SMCC_MRB). When FIFOs are enabled, bits I2SMCC_ISRA.TXLRDYx and I2SMCC_ISRA.TXRRDYx must not be used. Bits TXFFRDY, TXFFEMP, RXFFRDY and RXFFFUL of the Interrupt Status register B (I2SMCC_ISRB) register provide the status of the FIFOs. These status bits are only available if the FIFOEN bit is set.

Each I^2S or TDM channel has its own register. The register depth depending on the configuration is available in the following table.

Table 45-6. Registers Depth

I2SMCC_MRA.FORMAT	I2SMCC_MRA.WIRECFG	I2SMCC_MRA.NBCHAN	Register Depth
0 or 1 (I^2S or LJ)	0	NA	4 words
	1		2 words
	2		1 word
2 or 3 (TDM or TDMLJ)	NA	0 or 1	4 words
		2 or 3	2 words
		4, 5, 6 or 7	1 word

45.6.10.1 Dedicated Registers

When a new data word is available in I2SMCC_RHLxR, the corresponding I2SMCC_ISRA.RXLRDYx is set. Reading I2SMCC_RHLxR clears this bit. When a new data word is available in I2SMCC_RHRxR, the corresponding I2SMCC_ISRA.RXRRDYx is set. Reading I2SMCC_RHRxR clears this bit.

A receive overrun condition occurs if a new data word becomes available for the left channel x before the previous data word has been read from I2SMCC_RHLxR. In this case, the Receive Left x Overrun bit (RXLOVFx) in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit. A receive overrun condition occurs if a new data word becomes available for the right channel x before the previous data word has been read from I2SMCC_RHRxR. In this case, the Receive Right x Overrun bit (RXROVFx) in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

When I2SMCC_THLxR is empty, I2SMCC_ISRA.TXLRDYx is set. Writing to I2SMCC_THLxR clears this bit. When I2SMCC_THRxR is empty, I2SMCC_ISRA.TXRRDYx is set. Writing to I2SMCC_THRxR clears this bit.

A transmit underrun condition occurs if a new data word needs to be transmitted before it has been written to I2SMCC_THLxR. In this case, the Transmit Left x Underrun (TXLUNFx) bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit. A transmit underrun condition occurs if a new data word needs to be transmitted before it has been written to I2SMCC_THRxR. In this case, the Transmit Right x Underrun (TXRUNFx) bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

In case of transmit underrun, if I2SMCC_MRA.TXSAME is '0', then a zero data word is transmitted. If I2SMCC_MRA.TXSAME is '1', then the previous data word for the current transmit channel number is transmitted.

Data words are right-justified in all dedicated registers (I2SMCC_RHLxR, I2SMCC_RHRxR, I2SMCC_THLxR and I2SMCC_THRxR). The 16-bit compact and 8-bit compact stereo data formats are not supported through dedicated registers.

In TDM or TDM Left-Justified mode, the dedicated left registers are used for even-numbered channels and the dedicated right registers are used for odd-numbered channels. The data of TDM channel 0 can be read through the I2SMCC_RHL0R and written through the I2SMCC_THL0R, the data of TDM channel 1 can be read through the I2SMCC_RHR0R and written through the I2SMCC_THR0R. Data read/write continues in this manner up to the data of TDM channel 6 read through the I2SMCC_RHL3R and written through the I2SMCC_THL3R and the data of TDM channel 7 read through the I2SMCC_RHR3R and written through the I2SMCC_THR3R.

45.6.10.2 Common Registers

The Receiver Holding Register (I2SMCC_RHR) and the Transmitter Holding Register (I2SMCC_THR) provide an access to all channels enabled through a single location.

When a new data word is available, the corresponding RXLRDYx bit or RXRRDYx bit in I2SMCC_ISRA is set. Reading I2SMCC_RHR clears this bit. In I^{2}S or Left-Justified mode with the Common register Access Mode bit (CRAMODE) of the Mode register B (I2SMCC_MRB) set to '0', consecutive access to I2SMCC_RHR reads first all the left channels enabled then all the right channels enabled. In I^{2}S or Left-Justified mode with the I2SMCC_MRB.CRAMODE bit set to '1', consecutive access to I2SMCC_RHR reads first the left and right channels of the first , then the left and right channels of the second I^{2}S enabled and so on until the last I^{2}S enabled. In TDM or TDM Left-Justified mode, consecutive access to I2SMCC_RHR reads the TDM channels enabled.

A receive overrun condition occurs if a new data word becomes available for the left channel x or right channel x before the previous data word has been read from I2SMCC_RHR. In this case, the corresponding RXLOVFx or RXROVFx bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

When a data can be written in I2SMCC_THR, the corresponding TXLRDYx bit or TXRRDYx bit in I2SMCC_ISRA is set. Writing to I2SMCC_THR clears this bit. In I^{2}S or Left-Justified mode, with I2SMCC_MRB.CRAMODE set to '0', consecutive access to I2SMCC_THR writes first all the left channels enabled then all the right channels enabled. In I^{2}S or Left-Justified mode, with I2SMCC_MRB.CRAMODE set to '1', consecutive access to I2SMCC_THR writes first the left and right channels of the first I^{2}S enabled, then the left and right channels of the second I^{2}S enabled and continues in this manner until the last I^{2}S enabled. In TDM or TDM Left-Justified mode, consecutive access to I2SMCC_THR writes the TDM channels enabled.

A transmit underrun condition occurs if a new data word needs to be transmitted before it has been written to I2SMCC_THR. In this case, the corresponding TXLUNFx or TXRUNFx bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

In case of transmit underrun, if the value of I2SMCC_MRA.TXSAME is '0', then a '0' is transmitted. If the value of I2SMCC_MRA.TXSAME is '1', then the previous data word for the current transmit channel number is transmitted.

After a transmit underrun, the data written in I2SMCC_THR is discarded to keep the left and right channels synchronized.

Data words are right-justified in the common registers (I2SMCC_RHR and I2SMCC_THR). For the 16-bit compact stereo data format, the left sample uses bits [15:0] and the right sample uses bits [31:16] of the same data word. For the 8-bit compact stereo data format, the left sample uses bits [7:0] and the right sample uses bits [15:8] of the same data word.

45.6.11 DMA Controller Operation

All receiver audio channels and all transmitter audio channels are each assigned to a single DMA Controller channel.

The DMA Controller reads from the I2SMCC_RHR and writes to the I2SMCC_THR for all audio channels successively.

The DMA Controller transfers may use 32-bit word, 16-bit halfword, or 8-bit byte depending on the value of the I2SMCC_MRA.DATALENGTH field.

The DMA chunk size field (DMACHUNK) of the Mode Register B(I2SMCC_MRB) should correspond to the DMA channel configuration. The supported chunk according to the configuration is available in the two tables below

Table 45-7. TX DMA Chunk Configurations

<table>
<tr><th>FORMAT</th><th>TXMONO</th><th>DATALENGTH</th><th>NBCHAN</th><th>DMACHUNK</th><th>Description</th></tr>
<tr><td rowspan="12">0 or 1
(I²S or LJ)</td><td rowspan="4">0
(Stereo)</td><td rowspan="8">0, 1, 2, 3, 4 or 6
(32, 24, 20, 18, 16 or 8 bits)</td><td rowspan="12">NA</td><td>0</td><td>1-word chunk</td></tr>
<tr><td>1</td><td>2-word chunk</td></tr>
<tr><td>2</td><td>4-word chunk</td></tr>
<tr><td>3</td><td>8-word chunk</td></tr>
<tr><td rowspan="4">1
(Mono)</td><td>0</td><td>1-word chunk</td></tr>
<tr><td>1</td><td>2-word chunk</td></tr>
<tr><td>2</td><td rowspan="2">4-word chunk</td></tr>
<tr><td>3</td></tr>
<tr><td rowspan="4">0 or 1
(Stereo or Mono)</td><td rowspan="4">5 or 7
(16 bits compact or 8 bits compact)</td><td>0</td><td>1-word chunk</td></tr>
<tr><td>1</td><td>2-word chunk</td></tr>
<tr><td>2</td><td rowspan="2">4-word chunk</td></tr>
<tr><td>3</td></tr>
</table>

...........continued

FORMAT	TXMONO	DATALENGTH	NBCHAN	DMACHUNK	Description
2 or 3 (TDM or TDMLJ)	NA	NA	0 (1 channel)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	0 (Stereo)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	8-word chunk
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	1 (Mono)		1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	
				3	
	0 or 1 (Stereo or Mono)	5 or 7 (16 bits compact or 8 bits compact)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	
				3	

Table 45-8. RX DMA Chunk Configurations

FORMAT	DATALENGTH	NBCHAN	DMACHUNK	Description
0 or 1 (I^2S or LJ)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	NA	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	8-word chunk
	5 or 7 (16 bits compact or 8 bits compact)		0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
2 or 3 (TDM or TDMLJ)	NA	0 (1 channel)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	8-word chunk
		2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
	5 or 7 (16 bits compact or 8 bits compact)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
		2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
			1	2-word chunk
			2	
			3	

45.6.12 Loop-back Mode

For debug purposes, the I2SMCC can be configured to loop back the transmitter to the receiver. Writing a ’1’ to the I2SMCC_MRA.RXLOOP bit internally connects I2SMCC_DOUTx to I2SMCC_DINx, so that the transmitted data is also received. Writing a ’0’ to I2SMCC_MRA.RXLOOP restores the normal behavior with independent receiver and transmitter. As for other changes to the receiver or transmitter configuration, the I2SMCC receiver and transmitter must be disabled before writing to I2SMCC_MRA to update I2SMCC_MRA.RXLOOP.

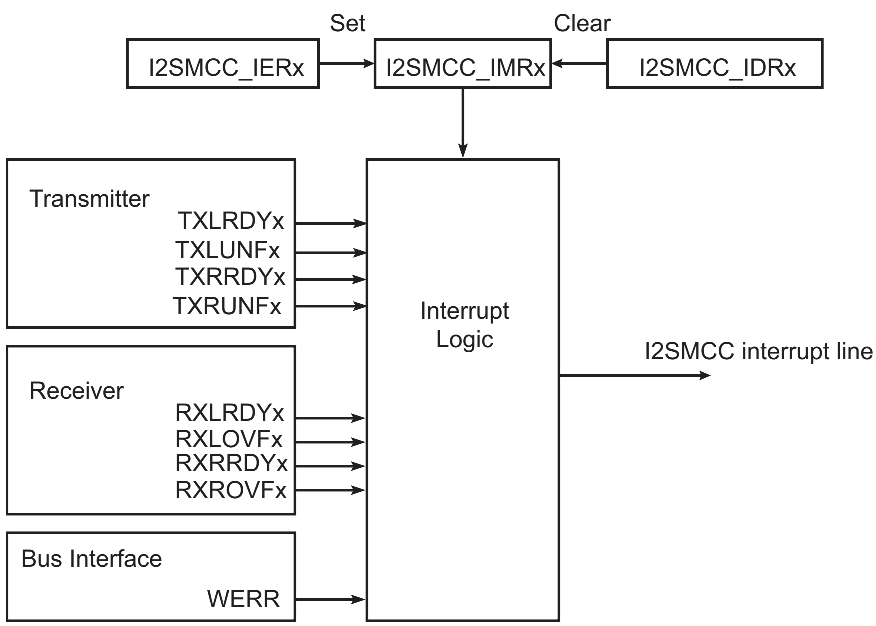
45.6.13 Interrupts

An I2SMCC interrupt request can be triggered whenever one or several of the following bits are set in I2SMCC_ISRA and/or I2SMCC_ISRB:

- Receive Left x Ready (RXLRDYx)
- Receive Right x Ready (RXRRDYx)
- Receive Left x Overrun (RXLOVFx)
- Receive Right x Overrun (RXROVFx)
- Transmit Left x Ready (TXLRDYx)
- Transmit Right x Ready (TXRRDYx)
- Transmit Left x Underrun (TXLUNFx)
- Transmit Right x Underrun (TXRUNFx)
- Write Error (WERR)
- Transmit FIFO Ready (TXFFRDY)
- Transmit FIFO Empty (TXFFEMP)
- Receive FIFO Ready (RXFFRDY)
- Receive FIFO Full (RXFFFUL)

The interrupt request is generated if the corresponding bit in the Interrupt Mask registers (I2SMCC_IMRA and I2SMCC_IMRB) is set. Bits in I2SMCC_IMRx are set by writing a '1' to the corresponding bit in I2SMCC_IERx and cleared by writing a '1' to the corresponding bit in I2SMCC_IDRx. The interrupt request remains active until the corresponding bit in I2SMCC_ISRx is cleared.

Figure 45-7. Interrupt Block Diagram



45.6.14 Register Write Protection

To prevent any single software error from corrupting I2SMCC behavior, certain registers in the address space can be write-protected by setting the Write Protection Configuration Enable (WPCFEN), Write Protection Interrupt Enable (WPITEN) and/or Write Protection Control Enable (WPCTEN) bit(s) in the Write Protection Mode register (I2SMCC_WPMR).

If a write access to the protected registers is detected, the Write Protection Violation Status (WPVS) flag in the Write Protection Status register (I2SMCC_WPSR) is set and the field Write Protection Violation Source (WPVSRC) indicates the register in which the write access has been attempted. An interrupt can be raised if the Write Error (WERR) interrupt is set in I2SMCC_IMRB.

The WPVS flag is automatically reset by reading I2SMCC_WPSR.

The following register can be write-protected with the I2SMCC_WPMR.WPCFEN bit:

- Inter-IC Sound Multi Channel Controller Mode Register A
- Inter-IC Sound Multi Channel Controller Mode Register B

The following registers can be write-protected with the I2SMCC_WPMR.WPITEN bit:

- Inter-IC Sound Multi Channel Controller Interrupt Enable Register A
- Inter-IC Sound Multi Channel Controller Interrupt Disable Register A
- Inter-IC Sound Multi Channel Controller Interrupt Enable Register B
- Inter-IC Sound Multi Channel Controller Interrupt Disable Register B

The following register can be write-protected with the I2SMCC_WPMR.WPCTEN bit:

- Inter-IC Sound Multi Channel Controller Control Register

45.6.15 Functional Safety (Protection, Monitors and Reports)

45.6.15.1 Protections

The configuration, interrupt and control registers can be protected against unintentional write accesses resulting from an erroneous software, bad DMA configuration or any abnormal single event upset that would create a spurious access on the bus. See Register Write Protection for a detailed description.

45.6.15.2 Monitors and Reports

When register write protection is enabled, any incorrect access is reported in I2SMCC_WPSR.WPVS and in I2SMCC_ISRB.WERR. It is possible to trigger an interrupt by writing a ‘1’ in I2SMCC_IERB.WERR.

I2SMCC embeds an on-the-fly monitoring of the WS and CK output pads to speed-up detection and report of any error while transmitting a data. The monitor cannot be disabled for safety reason and only reports an error in the user interface (i.e. there is no action on transmission path).

The I2SMCC internal outputs are passed through IO multiplexing logic that may be unintentionally badly configured and leads to absence or bad transmission. Other causes of stuck-at are detected (external cause such as short-circuits or internal cause such as pad buffer transistor failure). As an example, a badly re-assigned I2SMCC_WS/CK IO pin (software error) is detected as soon as the I2SMCC starts transmitting a data because it is unlikely for another peripheral sharing the same IO pin to drive the same waveform as the I2SMCC on I2SMCC_WS/CK output.

The monitor is enabled when the I2SMCC is configured in Host mode (I2SMCC_MRA.MODE=1).

An error is reported in I2SMCC_WPSR.PADERR and in I2SMCC_ISR.SECE when there is evidence that the output values on I2SMCC_WS or I2SMCC_CK differ from the internal values generated by the I2SMCC.

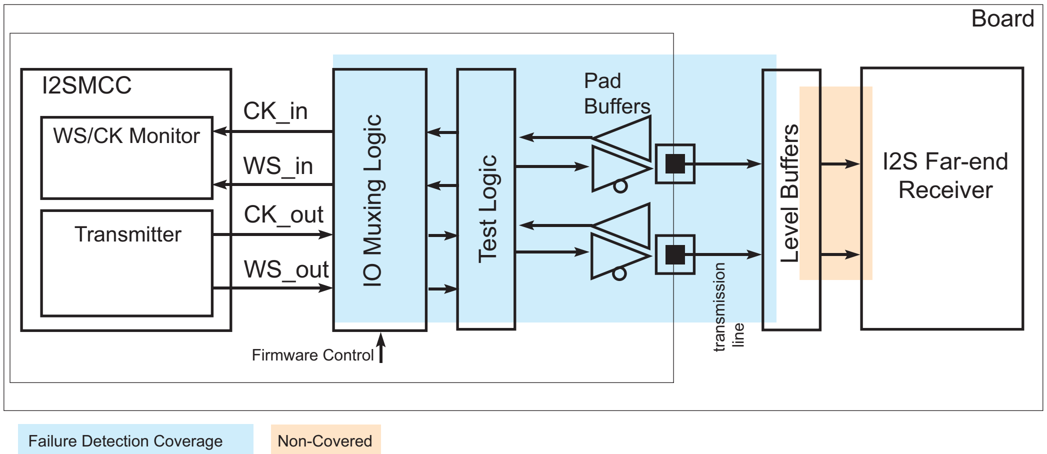
When an I2S format is configured (I2SMCC_MRA.FORMAT < 2), the I2SMCC_WS output pad value is oversampled by 2 and, as soon as 2 consecutive samples differ from the internal value generated by the I2SMCC, an error is reported. The rising edges of the I2SMCC_CK output are counted during a reference period equal to an audio sample period and at the end of each counting period, an error is reported if the counter is below seven at the end of the reference period (the minimum data length of an audio sample being eight bits).

When a TDM format is configured (I2SMCC_MRA.FORMAT > 1), the I2SMCC_WS output is monitored in a manner that differs from I2S formats because the waveform of WS can be configured with a minimum high duration pulse of 1 bit time. An error is reported if no falling edge is detected (thus the monitor checks that the output level is not stuck at 1 or 0). The I2SMCC_CK check is performed in the same way as in I2S format.

The detection method minimizes the likelihood to report a false positive that could result for example from a single upset event.

The monitor covers any failure that would be located in the I2SMCC IO multiplexing downstream circuitry, test logic, output pad buffer and external transmission line from the output pad to any buffering circuitry (if discrete components are placed between the transmitter and far-end receiver).

Figure 45-8. Failure Detection Coverage for the I2SMCC Monitor



When an error is reported, it is possible to trigger an interrupt by writing a ‘1’ in I2SMCC_IER.SECE.

45.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	I2SMCC_CR	31:24								
		23:16								
		15:8								
		7:0	SWRST		TXDIS	TXEN	CKDIS	CKEN	RXDIS	RXEN
0x04	I2SMCC_MRA	31:24	IWS	IMCKMODE	ISCKDIV[5:0]					
		23:16	TDMFS[1:0]		IMCKDIV[5:0]					
		15:8	NBCHAN[2:0]			SRCCLK	TXSAME	TXMONO	RXLOOP	RXMONO
		7:0	FORMAT[1:0]		WIRECFG[1:0]		DATALENGTH[2:0]			MODE
0x08	I2SMCC_MRB	31:24								
		23:16								
		15:8							DMACHUNK[1:0]	
		7:0				FIFOEN				CRAMODE
0x0C	I2SMCC_SR	31:24								
		23:16								
		15:8								
		7:0				TXEN				RXEN
0x10	I2SMCC_IERA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x14	I2SMCC_IDRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x18	I2SMCC_IMRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x1C	I2SMCC_ISRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x20	I2SMCC_IERB	31:24								
		23:16								
		15:8			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
		7:0								WERR
0x24	I2SMCC_IDRB	31:24								
		23:16								
		15:8			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
		7:0								WERR
0x28	I2SMCC_IMRB	31:24								
		23:16								
		15:8			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
		7:0								WERR
0x2C	I2SMCC_ISRB	31:24								
		23:16								
		15:8			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
		7:0								WERR
0x30	I2SMCC_RHR	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x34	I2SMCC_THR	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x38 ... 0x3F	Reserved									
0x40	I2SMCC_RHL0R	31:24	RHL[31:24]							
		23:16	RHL[23:16]							
		15:8	RHL[15:8]							
		7:0	RHL[7:0]							
0x44	I2SMCC_RHR0R	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x48	I2SMCC_RHL1R	31:24	RHL[31:24]							
		23:16	RHL[23:16]							
		15:8	RHL[15:8]							
		7:0	RHL[7:0]							
0x4C	I2SMCC_RHR1R	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x50	I2SMCC_RHL2R	31:24	RHL[31:24]							
		23:16	RHL[23:16]							
		15:8	RHL[15:8]							
		7:0	RHL[7:0]							
0x54	I2SMCC_RHR2R	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x58	I2SMCC_RHL3R	31:24	RHL[31:24]							
		23:16	RHL[23:16]							
		15:8	RHL[15:8]							
		7:0	RHL[7:0]							
0x5C	I2SMCC_RHR3R	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x60	I2SMCC_THL0R	31:24	THL[31:24]							
		23:16	THL[23:16]							
		15:8	THL[15:8]							
		7:0	THL[7:0]							
0x64	I2SMCC_THR0R	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							
0x68	I2SMCC_THL1R	31:24	THL[31:24]							
		23:16	THL[23:16]							
		15:8	THL[15:8]							
		7:0	THL[7:0]							
0x6C	I2SMCC_THR1R	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							
0x70	I2SMCC_THL2R	31:24	THL[31:24]							
		23:16	THL[23:16]							
		15:8	THL[15:8]							
		7:0	THL[7:0]							
0x74	I2SMCC_THR2R	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	I2SMCC_THL3R	31:24	THL[31:24]							
		23:16	THL[23:16]							
		15:8	THL[15:8]							
		7:0	THL[7:0]							
0x7C	I2SMCC_THR3R	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							
0x80 ... 0xE3	Reserved									
0xE4	I2SMCC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCTEN	WPITEN	WPCFEN
0xE8	I2SMCC_WPSR	31:24	WPVSRC[23:16]							
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

45.7.1 I2SMCC Control Register

Name: I2SMCC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SWRST		TXDIS	TXEN	CKDIS	CKEN	RXDIS	RXEN
Access	W		W	W	W	W	W	W
Reset	–		–	–	–	–	–	–

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets all the registers in the I2SMCC. The I2SMCC is disabled after the reset.

Bit 5 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	Disables the I2SMCC transmitter. Bit I2SMCC_SR.TXEN is cleared when the Transmitter is stopped.

Bit 4 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	Enables the I2SMCC transmitter, if TXDIS is not ‘1’. Bit I2SMCC_SR.TXEN is set when the Transmitter is started.

Bit 3 – CKDIS Clocks Disable

Value	Description
0	No effect.
1	Disables the I2SMCC clock generation.

Bit 2 – CKEN Clocks Enable

Value	Description
0	No effect.
1	Enables the I2SMCC clock generation, if CKDIS is not ‘1’.

Bit 1 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	Disables the I2SMCC receiver. Bit I2SMCC_SR.RXEN is cleared when the receiver is stopped.

Bit 0 – RXEN Receiver Enable

Value	Description
0	No effect.
1	Enables the I2SMCC receiver, if RXDIS is not ‘1’. Bit I2SMCC_SR.RXEN is set when the receiver is activated.

45.7.2 I2SMCC Mode Register A

Name: I2SMCC_MRA
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in Inter-IC Sound Write Protection Mode Register.

The I2SMCC_MRA must only be written when the I2SMCC is stopped in order to avoid unexpected behavior on the I2SMCC_WS, I2SMCC_CK and I2SMCC_DOUTx outputs. The proper sequence is to write to I2SMCC_MRA, then write to I2SMCC_CR to enable the I2SMCC or to disable the I2SMCC before writing a new value to I2SMCC_MRA.

Bit	31	30	29	28	27	26	25	24
	IWS	IMCKMODE	ISCKDIV[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TDMFS[1:0]		IMCKDIV[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NBCHAN[2:0]			SRCCLK	TXSAME	TXMONO	RXLOOP	RXMONO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FORMAT[1:0]		WIRECFG[1:0]		DATALENGTH[2:0]			MODE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – IWS I2SMCC_WS Slot Length
See tables Slot Length (I^{2}S format) and Slot Length (Left-Justified format).

Value	Description
0	I2SMCC_WS slot is 32 bits long for DATALENGTH = 18/20/24 bits.
1	I2SMCC_WS slot is 24 bits long for DATALENGTH = 18/20/24 bits.

Bit 30 – IMCKMODE Host Clock Mode

Value	Description
0	No host clock generated.
1	Host clock generated.

Bits 29:24 – ISCKDIV[5:0] Selected Clock to I2SMCC Serial Clock Ratio
I2SMCC_CK Serial clock output frequency is Selected Clock divided by (2 * ISCKDIV). If ISCKDIV is 0, the I2SMCC_CK Serial clock output frequency is equal to the Selected Clock frequency.

Bits 23:22 – TDMFS[1:0] TDM Frame Synchronization

Value	Name	Description
0	SLOT	I2SMCC_WS pulse is high for one time slot at beginning of frame.
1	HALF	I2SMCC_WS pulse is high for half the time slots at beginning of frame.
2	BIT	I2SMCC_WS pulse is high for one bit period at beginning of frame, i.e., one ISCK period.

Bits 21:16 – IMCKDIV[5:0] Selected Clock to I2SMCC Host Clock Ratio
I2SMCC_MCK Host clock output frequency is Selected Clock divided by (2 * IMCKDIV). If IMCKDIV is 0, the I2SMCC_MCK Host clock output frequency is equal to the Selected Clock frequency.

Bits 15:13 – NBCHAN[2:0] Number of TDM Channels-1
Must be written with the number of TDM channels minus one.

Bit 12 – SRCCLK Source Clock Selection

Value	Description
0	The Peripheral clock is selected as source clock for I2SMCC_MCK/WS/CK pins.
1	The PMC.GCLKx clock is selected as source clock (I2SMCC_MCK/WS/CK rate can be independent of system bus clock).

Bit 11 – TXSAME Transmit Data when Underrun

Value	Description
0	‘0’ is transmitted when underrun.
1	Previous sample transmitted when underrun.

Bit 10 – TXMONO Transmit Mono

Value	Description
0	Stereo
1	Mono, with left audio samples duplicated to right audio channel by the I2SMCC.

Bit 9 – RXLOOP Loop-back Test Mode

Value	Description
0	Normal mode
1	I2SMCC_DOUT outputs of I2SMCC are internally connected to I2SMCC_DIN inputs.

Bit 8 – RXMONO Receive Mono

Value	Description
0	Stereo
1	Mono, with left audio samples duplicated to right audio channel by the I2SMCC.

Bits 7:6 – FORMAT[1:0] Data Format

Value	Name	Description
0	I2S	I^2S format, stereo with I2SMCC_WS low for left channel, and MSB of sample starting one I2SMCC_CK period after I2SMCC_WS edge.
1	LJ	Left-justified format, stereo with I2SMCC_WS high for left channel, and MSB of sample starting on I2SMCC_WS edge.
2	TDM	TDM format, with (NBCHAN + 1) channels, I2SMCC_WS high at beginning of first channel, and MSB of sample starting one I2SMCC_CK period after I2SMCC_WS edge.
3	TDMLJ	TDM format, left-justified, with (NBCHAN + 1) channels, I2SMCC_WS high at beginning of first channel, and MSB of sample starting on I2SMCC_WS edge.

Bits 5:4 – WIRECFG[1:0] Wire Configuration

Value	Name	Description
0	I2S_1_TDM_0	In I^2S and LJ formats, I2SMCC_DIN0 and I2SMCC_DOUT0 are used to transmit and receive I^2S frames. In TDM and TDMLJ formats, I2SMCC_DIN0 and I2SMCC_DOUT0 are used to transmit and receive TDM frames.
1	I2S_2_TDM_1	In I^2S and LJ formats, I2SMCC_DIN1..0 and I2SMCC_DOUT1..0 are used to transmit and receive 2 I^2S frames. In TDM and TDMLJ formats, I2SMCC_DIN1 and I2SMCC_DOUT1 are used to transmit and receive TDM frames.
2	I2S_4_TDM_2	In I^2S and LJ formats, I2SMCC_DIN3..0 and I2SMCC_DOUT3..0 are used to transmit and receive 4 I^2S frames. In TDM and TDMLJ formats, I2SMCC_DIN2 and I2SMCC_DOUT2 are used to transmit and receive TDM frames.

Value	Name	Description
3	TDM_3	In I^2S and LJ formats, reserved for future use, do not use. In TDM and TDMLJ formats, I2SMCC_DIN3 and I2SMCC_DOUT3 are used to transmit and receive TDM frames.

Bits 3:1 – DATALENGTH[2:0] Data Word Length

Value	Name	Description
0	32_BITS	Data length is set to 32 bits.
1	24_BITS	Data length is set to 24 bits.
2	20_BITS	Data length is set to 20 bits.
3	18_BITS	Data length is set to 18 bits.
4	16_BITS	Data length is set to 16 bits.
5	16_BITS_COMPACT	Data length is set to 16-bit compact stereo. Left sample in bits [15:0] and right sample in bits [31:16] of same word.
6	8_BITS	Data length is set to 8 bits.
7	8_BITS_COMPACT	Data length is set to 8-bit compact stereo. Left sample in bits [7:0] and right sample in bits [15:8] of the same word.

Bit 0 – MODE I2SMCC Mode

Value	Name	Description
0	SLAVE	Client mode. I2SMCC_CK and I2SMCC_WS pin inputs used as bit clock and word select/ frame synchronization.
1	MASTER	Host mode. Bit clock and word select/frame synchronization generated by I2SMCC from Peripheral Clock or GCLK if I2SMCC_MCK/WS/CK rates must be independent of system bus clock (See I2SMCC_MR.SRCCLK) and output to I2SMCC_CK and I2SMCC_WS pins. MCK is output as host clock on I2SMCC_MCK if I2SMCC_MRA.IMCKMODE is set.

45.7.3 I2SMCC Mode Register B

Name: I2SMCC_MRB
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The I2SMCC_MRB must only be written when the I2SMCC is stopped in order to avoid unexpected behavior on the I2SMCC_WS, I2SMCC_CK and I2SMCC_DOUT outputs. The proper sequence is to write to I2SMCC_MRB, then write to I2SMCC_CR to enable the I2SMCC or to disable the I2SMCC before writing a new value to I2SMCC_MRB.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DMACHUNK[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
				FIFOEN				CRAMODE
Access				R/W				R/W
Reset				0				0

Bits 9:8 – DMACHUNK[1:0] DMA Chunk Size

Value	Name	Description
0	1_WORD	Each DMA transfer contains 1 word.
1	2_WORDS	Each DMA transfer contains 2 words.
2	4_WORDS	Each DMA transfer contains 4 words.
3	8_WORDS	Each DMA transfer contains 8 words.

Bit 4 – FIFOEN FIFO Enable

Value	Description
0	The Receive and Transmit FIFOs are disabled.
1	The Receive and Transmit FIFOs are enabled. Transmit data can only be written through I2SMCC_THR. Receive data can only be read through I2SMCC_RHR.

Bit 0 – CRAMODE Common Register Access Mode

Value	Name	Description
0	LEFT_FIRST	All enabled I^2S left channels are filled first, then I^2S right channels.
1	REGULAR	An enabled I^2S left channel is filled, then the corresponding right channel, until all channels are filled.

45.7.4 I2SMCC Status Register

Name: I2SMCC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				TXEN				RXEN
Access				R				R
Reset				0				0

Bit 4 – TXEN Transmitter Enabled
TXEN=1 if the selected clock (internal bit clock, see the figure I2SMCC Clock Generation) is active and WS is active.

Value	Description
0	Cleared when the transmitter is disabled, following a I2SMCC_CR.TXDIS or I2SMCC_CR.SWRST request.
1	Set when the transmitter is enabled, following a I2SMCC_CR.TXEN request.

Bit 0 – RXEN Receiver Enabled
RXEN=1 if the selected clock (internal bit clock, see the figure I2SMCC Clock Generation) is active and WS is active.

Value	Description
0	Cleared when the receiver is disabled, following an RXDIS or SWRST request in I2SMCC_CR.
1	Set when the receiver is enabled, following an RXEN request in I2SMCC_CR.

45.7.5 I2SMCC Interrupt Enable Register A

Name: I2SMCC_IERA
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 25, 27, 29, 31 – RXROVFx I^{2}S Receive Right x or TDM Channel [2x]+1 Overrun Interrupt Enable

Bits 24, 26, 28, 30 – RXLOVFx I^{2}S Receive Left x or TDM Channel 2x Overrun Interrupt Enable

Bits 17, 19, 21, 23 – RXRRDYx I^{2}S Receive Right x or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 16, 18, 20, 22 – RXLRDYx I^{2}S Receive Left x or TDM Channel 2x Ready Interrupt Enable

Bits 9, 11, 13, 15 – TXRUNFx I^{2}S Transmit Right x or TDM Channel [2x]+1 Underrun Interrupt Enable

Bits 8, 10, 12, 14 – TXLUNFx I^{2}S Transmit Left x or TDM Channel 2x Underrun Interrupt Enable

Bits 1, 3, 5, 7 – TXRRDYx I^{2}S Transmit Right x or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 0, 2, 4, 6 – TXLRDYx I^{2}S Transmit Left x or TDM Channel 2x Ready Interrupt Enable

45.7.6 I2SMCC Interrupt Disable Register A

Name: I2SMCC_IDRA
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 25, 27, 29, 31 – RXROVFx I^{2}S Receive Right x or TDM Channel [2x]+1 Overrun Interrupt Disable

Bits 24, 26, 28, 30 – RXLOVFx I^{2}S Receive Left x or TDM Channel 2x Overrun Interrupt Disable

Bits 17, 19, 21, 23 – RXRRDYx I^{2}S Receive Right x or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 16, 18, 20, 22 – RXLRDYx I^{2}S Receive Left x or TDM Channel 2x Ready Interrupt Disable

Bits 9, 11, 13, 15 – TXRUNFx I^{2}S Transmit Right x or TDM Channel [2x]+1 Underrun Interrupt Disable

Bits 8, 10, 12, 14 – TXLUNFx I^{2}S Transmit Left x or TDM Channel 2x Underrun Interrupt Disable

Bits 1, 3, 5, 7 – TXRRDYx I^{2}S Transmit Right x or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 0, 2, 4, 6 – TXLRDYx I^{2}S Transmit Left x or TDM Channel 2x Ready Interrupt Disable

45.7.7 I2SMCC Interrupt Mask Register A

Name: I2SMCC_IMRA
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding source of interrupt is disabled.

1: The corresponding source of interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 25, 27, 29, 31 – RXROVFx I^2S Receive Right x or TDM Channel [2x]+1 Overrun Interrupt Mask

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x or TDM Channel 2x Overrun Interrupt Mask

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x or TDM Channel 2x Ready Interrupt Mask

Bits 9, 11, 13, 15 – TXRUNFx I^2S Transmit Right x or TDM Channel [2x]+1 Underrun Interrupt Mask

Bits 8, 10, 12, 14 – TXLUNFx I^2S Transmit Left x or TDM Channel 2x Underrun Interrupt Mask

Bits 1, 3, 5, 7 – TXRRDYx I^2S Transmit Right x or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 0, 2, 4, 6 – TXLRDYx I^2S Transmit Left x or TDM Channel 2x Ready Interrupt Mask

45.7.8 I2SMCC Interrupt Status Register A

Name: I2SMCC_ISRA
Offset: 0x1C
Reset: 0x00000003
Property: Read-only

When I2SMCC_MRB.FIFOEN = 1, RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx are not relevant. Refer to I2SMCC_ISRB.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	1

Bits 25, 27, 29, 31 – RXROVFx I^2S Receive Right x or TDM Channel [2x]+1 Overrun Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an overrun error occurs in either I2SMCC_RHRxR or I2SMCC_RHR.

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x or TDM Channel 2x Overrun Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an overrun error occurs in either I2SMCC_RHLxR or I2SMCC_RHR.

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x or TDM Channel [2x]+1 Ready Flag (Cleared by reading I2SMCC_RHR/RHLxR)

Value	Description
0	Cleared when a predefined number of read accesses are performed in I2SMCC_RHRxR or I2SMCC_RHR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG / FORMAT and varies from 1 to 8.
1	Set when received data is available in either I2SMCC_RHRxR or I2SMCC_RHR.

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x or TDM Channel 2x Ready Flag (Cleared by reading I2SMCC_RHR/RHLxR)

Value	Description
0	Cleared when a predefined number of write accesses is performed in either I2SMCC_RHLxR or I2SMCC_RHR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG/ FORMAT and varies from 1 to 7.
1	Set when received data is available in either I2SMCC_RHLxR or I2SMCC_RHR.

Bits 9, 11, 13, 15 – TXRUNFx I^2S Transmit Right x or TDM Channel [2x]+1 Underrun Flag (Cleared on read)

Value	Description
0	Cleared when the I2SMCC_ISRA is read.
1	Set when an underrun error occurs in either I2SMCC_THR or I2SMCC_THRxR.

Bits 8, 10, 12, 14 – TXLUNFx I^2S Transmit Left x or TDM Channel 2x Underrun (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an underrun error occurs in either I2SMCC_THR or I2SMCC_THLxR.

Bits 1, 3, 5, 7 – TXRRDYx I^2S Transmit Right x or TDM Channel [2x]+1 Ready Flag (Cleared by writing I2SMCC_THR/THRxR)

Value	Description
0	Cleared when a predefined number of write accesses is performed in either I2SMCC_THRxR or I2SMCC_THR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG/FORMAT and varies from 1 to 7.
1	Set when I2SMCC_THR or I2SMCC_THRxR is empty.

Bits 0, 2, 4, 6 – TXLRDYx I^2S Transmit Left x or TDM Channel 2x Ready Flag (Cleared by writing I2SMCC_THR/THLxR)

Value	Description
0	Cleared when a predefined number of write accesses is performed in either I2SMCC_THLxR or I2SMCC_THR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG/FORMAT and varies from 1 to 7.
1	Set when I2SMCC_THR or I2SMCC_THLxR is empty.

45.7.9 I2SMCC Interrupt Enable Register B

Name: I2SMCC_IERB
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bits of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit	7	6	5	4	3	2	1	0
								WERR
Access								W
Reset								–

Bit 13 – RXFFFUL RX FIFO Full Interrupt Enable

Bit 12 – RXFFRDY RX FIFO Ready Interrupt Enable

Bit 9 – TXFFEMP TX FIFO Empty Interrupt Enable

Bit 8 – TXFFRDY TX FIFO Ready Interrupt Enable

Bit 0 – WERR Write Error Interrupt Enable

45.7.10 I2SMCC Interrupt Disable Register B

Name: I2SMCC_IDRB
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bits of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit	7	6	5	4	3	2	1	0
								WERR
Access								W
Reset								–

Bit 13 – RXFFFUL RX FIFO Full Interrupt Disable

Bit 12 – RXFFRDY RX FIFO Ready Interrupt Disable

Bit 9 – TXFFEMP TX FIFO Empty Interrupt Disable

Bit 8 – TXFFRDY TX FIFO Ready Interrupt Disable

Bit 0 – WERR Write Error Interrupt Disable

45.7.11 I2SMCC Interrupt Mask Register B

Name: I2SMCC_IMRB
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for the listed bits of this register:

0: The corresponding source of interrupt is disabled.

1: The corresponding source of interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
								WERR
Access								R
Reset								0

Bit 13 – RXFFFUL RX FIFO Full Interrupt Mask

Bit 12 – RXFFRDY RX FIFO Ready Interrupt Mask

Bit 9 – TXFFEMP TX FIFO Empty Interrupt Mask

Bit 8 – TXFFRDY TX FIFO Ready Interrupt Mask

Bit 0 – WERR Write Error Interrupt Mask

45.7.12 I2SMCC Interrupt Status Register B

Name: I2SMCC_ISRB
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

When I2SMCC_MRB.FIFOEN = 0, TXFFRDY, TXFFEMP, RXFFRDY and RXFFFUL are not relevant. Refer to I2SMCC_ISRA.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RXFFFUL	RXFFRDY			TXFFEMP	TXFFRDY
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
								WERR
Access								R
Reset								0

Bit 13 – RXFFFUL RX FIFO Full Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRB is read.
1	Set when RX FIFO is full and I2SMCC_MRB.FIFOEN = 1.

Bit 12 – RXFFRDY RX FIFO Ready Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRB is read.
1	Set when RX FIFO is ready to be read and I2SMCC_MRB.FIFOEN = 1.

Bit 9 – TXFFEMP TX FIFO Empty Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRB is read.
1	Set when TX FIFO is empty and I2SMCC_MRB.FIFOEN = 1.

Bit 8 – TXFFRDY TX FIFO Ready Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRB is read.
1	Set when TX FIFO is ready to be written and I2SMCC_MRB.FIFOEN = 1.

Bit 0 – WERR Write Error Flag (Cleared on read)

Value	Description
0	Cleared when the I2SMCC_ISRB is read.
1	Set when a write occurs in a protected register.

45.7.13 I2SMCC Receiver Holding Register

Name: I2SMCC_RHR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RHR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RHR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RHR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RHR[31:0] Receiver Holding Register
Set by hardware to the last received data word. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right justified in the RHR field.

45.7.14 I2SMCC Transmitter Holding Register

Name: I2SMCC_THR
Offset: 0x34
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	THR[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	THR[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	THR[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	THR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – THR[31:0] Transmitter Holding Register
Next data word to be transmitted after the current word if TXLRDYx or TXRRDYx is not set. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the THR field.

45.7.15 I2SMCC Receiver Holding Left x Register

Name: I2SMCC_RHLxR
Offset: 0x40 + x*0x08 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RHL[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RHL[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RHL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RHL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RHL[31:0] Receiver Holding Left
Set by hardware to either the last received data word of the left channel on wire x in I^2S mode (I2SMCC_MRA.FORMAT = ‘0’) or to the last received data word of the x^{th} TDM channel multiplied by 2. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the RHL field.

45.7.16 I2SMCC Receiver Holding Right x Register

Name: I2SMCC_RHRxR
Offset: 0x44 + x*0x08 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RHR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RHR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RHR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RHR[31:0] Receiver Holding Right
Set by hardware to either the last received data word of the right channel on wire x in I^2S mode (I2SMCC_MRA.FORMAT = '0') or to the last received data word of the x^{th} +1 TDM channel multiplied by 2. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the RHR field.

45.7.17 I2SMCC Transmitter Holding Left x Register

Name: I2SMCC_THLxR
Offset: 0x60 + x*0x08 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	THL[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	THL[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	THL[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	THL[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – THL[31:0] Transmitter Holding Left
Next data word to be transmitted on the left channel on wire x in I^2S mode or the x^{th} TDM channel multiplied by 2 after the current word if TXLRDYx is not set. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the THL field.

45.7.18 I2SMCC Transmitter Holding Right x Register

Name: I2SMCC_THRxR
Offset: 0x64 + x*0x08 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	THR[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	THR[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	THR[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	THR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – THR[31:0] Transmitter Holding Right
Next data word to be transmitted on the right channel on wire x in I^2S mode or the x^{th} + 1 TDM channel multiplied by 2 after the current word if TXRRDYx is not set. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the THR field.

45.7.19 I2SMCC Write Protection Mode Register

Name: I2SMCC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCTEN	WPITEN	WPCFEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x493253	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 2 – WPCTEN Write Protection Control Enable

Value	Description
0	Disables the write protection of the control if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the control if WPKEY matches to 0x493253 (I2S in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection of the interruption if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the interruption if WPKEY matches to 0x493253 (I2S in ASCII).

Bit 0 – WPCFEN Write Protection Configuration Enable

Value	Description
0	Disables the write protection of the configuration if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the configuration if WPKEY matches to 0x493253 (I2S in ASCII).

45.7.20 I2SMCC Write Protection Status Register

Name: I2SMCC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the I2SMCC_WPSR.
1	A write protection violation has occurred since the last read of the I2SMCC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

46. Synchronous Serial Controller (SSC)

46.1 Description

The Synchronous Serial Controller (SSC) provides a synchronous communication link with external devices. It supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync, etc.

The SSC contains an independent receiver and transmitter and a common clock divider. The receiver and the transmitter each interface with three signals: the TD/RD signal for data, the TK/RK signal for the clock and the TF/RF signal for the Frame Sync. The transfers can be programmed to start automatically or on different events detected on the Frame Sync signal.

The SSC high-level of programmability and its use of DMA enable a continuous high bit rate data transfer without processor intervention.

Featuring connection to the DMA, the SSC enables interfacing with low processor overhead to:

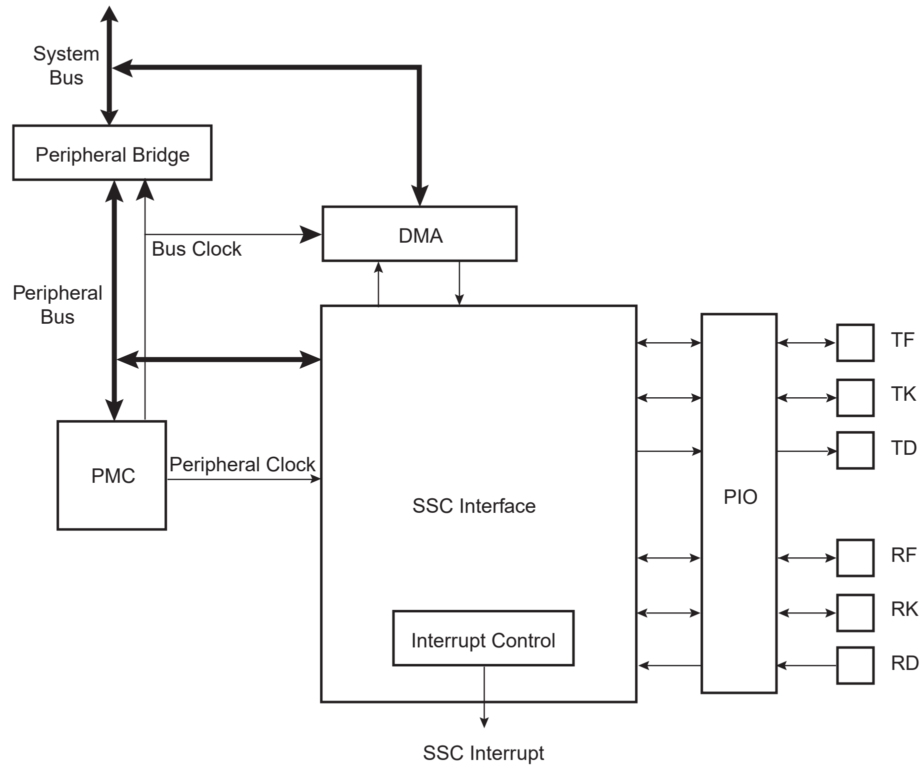
- Codecs in Host or Client mode
- DAC through dedicated serial interface, particularly I2S
- Magnetic card reader

46.2 Embedded Characteristics

- Provides Serial Synchronous Communication Links Used in Audio and Telecom Applications
- Contains an Independent Receiver and Transmitter and a Common Clock Divider
- Interfaced with the DMA Controller (DMAC) to Reduce Processor Overhead
- Offers a Configurable Frame Sync and Data Length
- Receiver and Transmitter Can be Programmed to Start Automatically or on Detection of Different Events on the Frame Sync Signal
- Receiver and Transmitter Include a Data Signal, a Clock Signal and a Frame Sync Signal
- 8 data transmit FIFO
- 8 data transmit FIFO

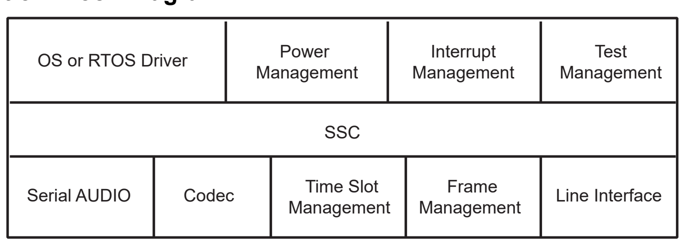
46.3 Block Diagram

Figure 46-1. SSC Block Diagram



46.4 Application Block Diagram

Figure 46-2. SSC Application Block Diagram



46.5 SSC Application Examples

The SSC can support several serial communication modes used in audio or high speed serial links. Some standard applications are shown in the following figures. All serial link applications supported by the SSC are not listed here.

Figure 46-3. Audio Application Block Diagram

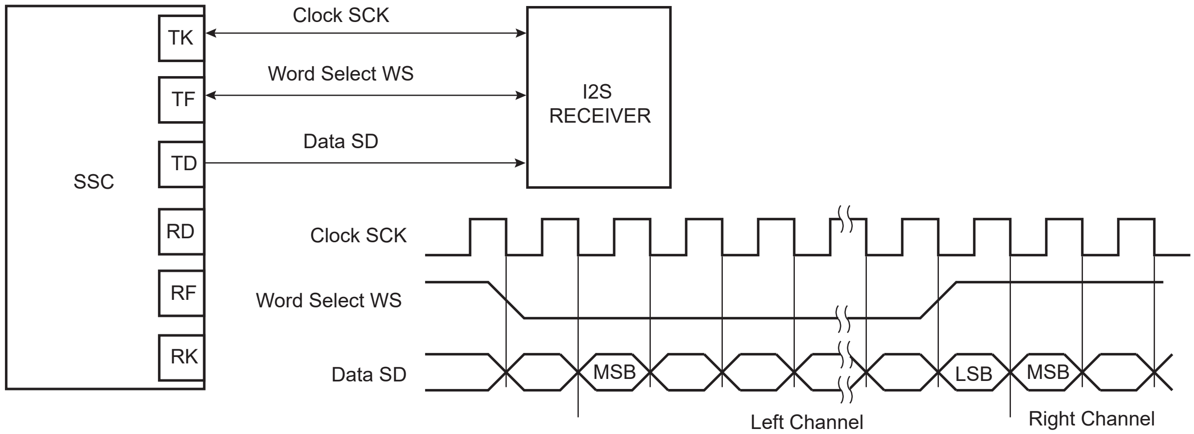


Figure 46-4. Codec Application Block Diagram

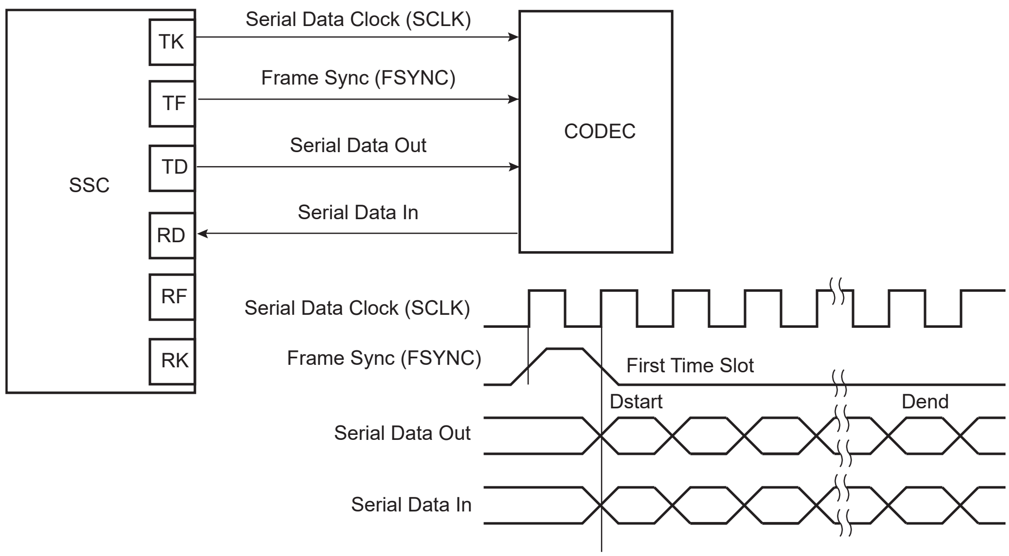
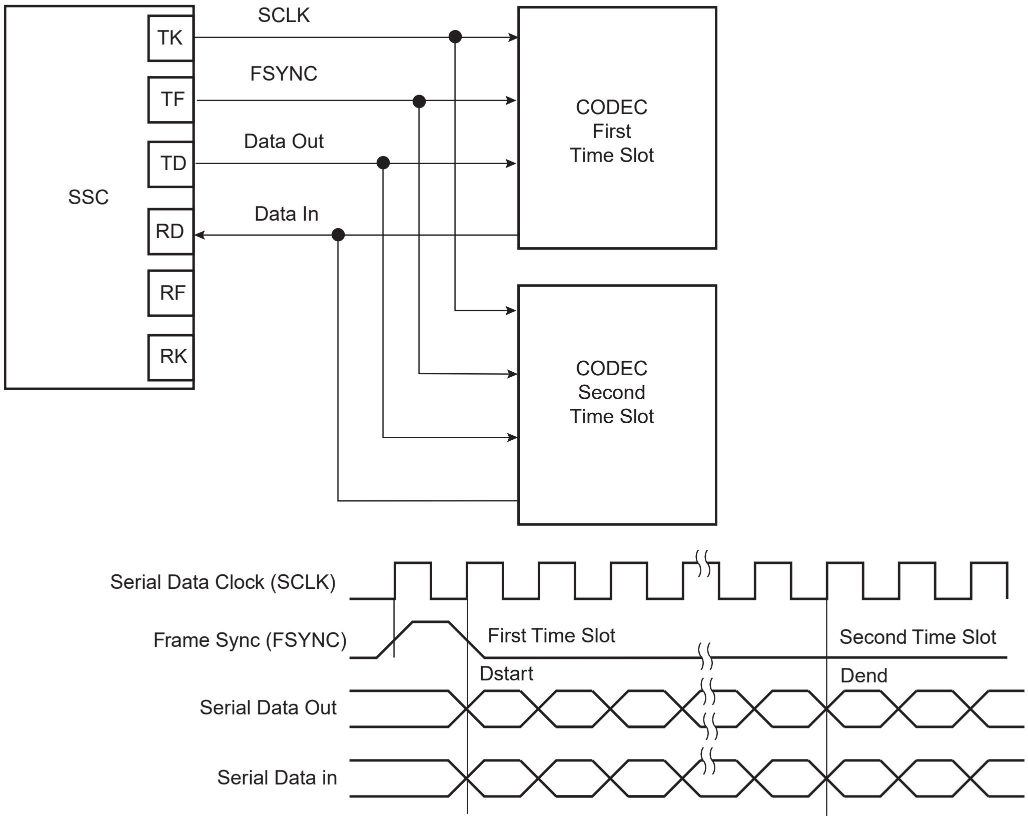


Figure 46-5. Time Slot Application Block Diagram



46.6 Pin Name List

Table 46-1. I/O Lines Description

Pin Name	Pin Description	Type
RF	Receive Frame Synchronization	Input/Output
RK	Receive Clock	Input/Output
RD	Receive Data	Input
TF	Transmit Frame Synchronization	Input/Output
TK	Transmit Clock	Input/Output
TD	Transmit Data	Output

46.7 Product Dependencies

46.7.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines.

Before using the SSC receiver, the PIO controller must be configured to dedicate the SSC receiver I/O lines to the SSC Peripheral mode.

Before using the SSC transmitter, the PIO controller must be configured to dedicate the SSC transmitter I/O lines to the SSC Peripheral mode.

46.7.2 Power Management

The SSC is not continuously clocked. The SSC interface may be clocked through the Power Management Controller (PMC), therefore the programmer must first configure the PMC to enable the SSC clock.

46.7.3 Interrupt

The SSC interface has an interrupt line connected to the interrupt controller. Handling interrupts requires programming the interrupt controller before configuring the SSC.

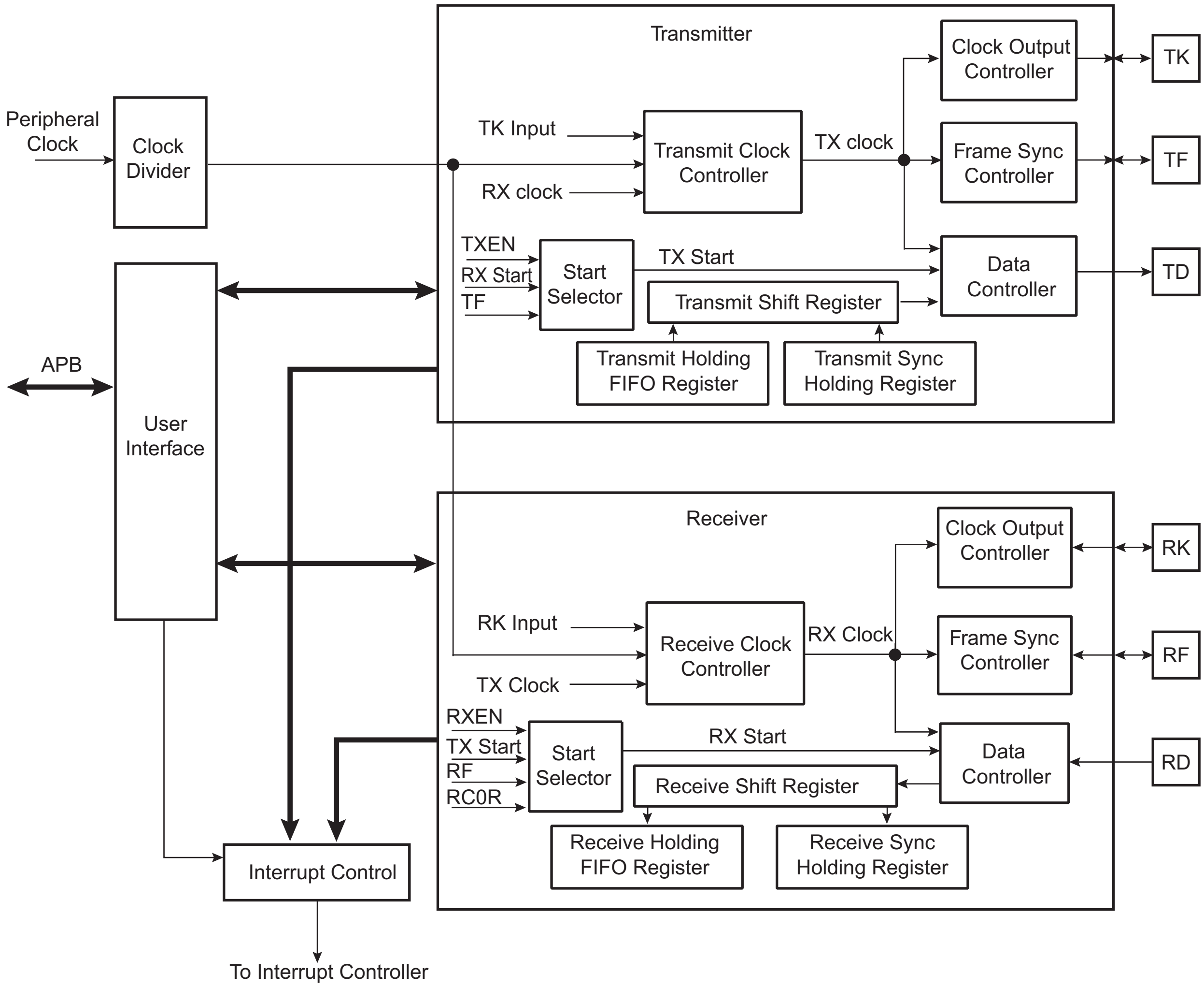
All SSC interrupts can be enabled/disabled configuring the SSC Interrupt Mask Register. Each pending and unmasked SSC interrupt asserts the SSC interrupt line. The SSC interrupt service routine can get the interrupt origin by reading the SSC Interrupt Status Register.

46.8 Functional Description

This section contains the functional description of the following: SSC Functional Block, Clock Management, Data Format, Start, Transmit, Receive and Frame Synchronization.

The receiver and transmitter operate separately. However, they can work synchronously by programming the receiver to use the transmit clock and/or to start a data transfer when transmission starts. Alternatively, this can be done by programming the transmitter to use the receive clock and/or to start a data transfer when reception starts. The transmitter and the receiver can be programmed to operate with the clock signals provided on either the TK or RK pins. This allows the SSC to support many Client mode data transfers. The maximum clock speed allowed on the TK and RK pins is the peripheral clock divided by 2.

Figure 46-6. SSC Functional Block Diagram



46.8.1 Clock Management

The transmit clock can be generated by:

- an external clock received on the TK I/O pad
- the receive clock
- the internal clock divider

The receive clock can be generated by:

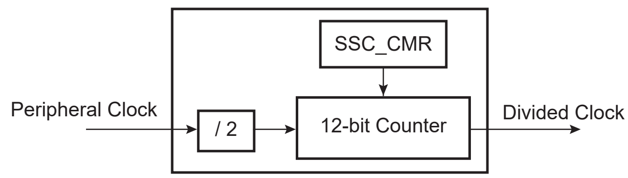
- an external clock received on the RK I/O pad
- the transmit clock
- the internal clock divider

Furthermore, the transmitter block can generate an external clock on the TK I/O pad, and the receive block can generate an external clock on the RK I/O pad.

This allows the SSC to support many Host and Client mode data transfers.

46.8.1.1 Clock Divider

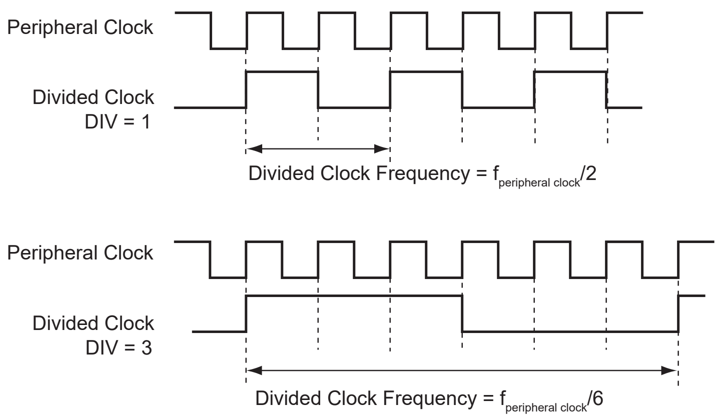
Figure 46-7. Divided Clock Block Diagram



The peripheral clock divider is determined by the 12-bit field DIV counter and comparator (so its maximal value is 4095) in the Clock Mode Register (SSC_CMR), allowing a peripheral clock division by up to 8190. The Divided Clock is provided to both the receiver and the transmitter. When this field is programmed to 0, the Clock Divider is not used and remains inactive.

When DIV is set to a value equal to or greater than 1, the Divided Clock has a frequency of peripheral clock divided by 2 times DIV. Each level of the Divided Clock has a duration of the peripheral clock multiplied by DIV. This ensures a 50% duty cycle for the Divided Clock regardless of whether the DIV value is even or odd.

Figure 46-8. Divided Clock Generation

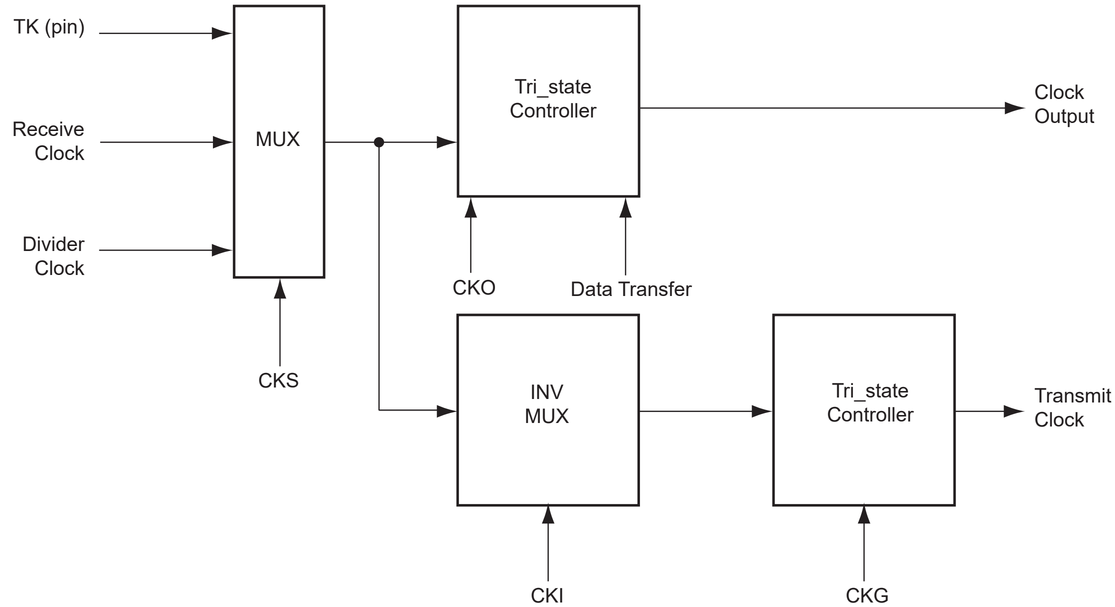


46.8.1.2 Transmit Clock Management

The transmit clock is generated from the receive clock or the divider clock or an external clock scanned on the TK I/O pad. The transmit clock is selected by the CKS field in the Transmit Clock Mode Register (SSC_TCMR). Transmit Clock can be inverted independently by the CKI bits in the SSC_TCMR.

The transmitter can also drive the TK I/O pad continuously or be limited to the current data transfer. The clock output is configured by the SSC_TCMR. The Transmit Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_TCMR to select TK pin (CKS field) and at the same time Continuous Transmit Clock (CKO field) can lead to unpredictable results.

Figure 46-9. Transmit Clock Management

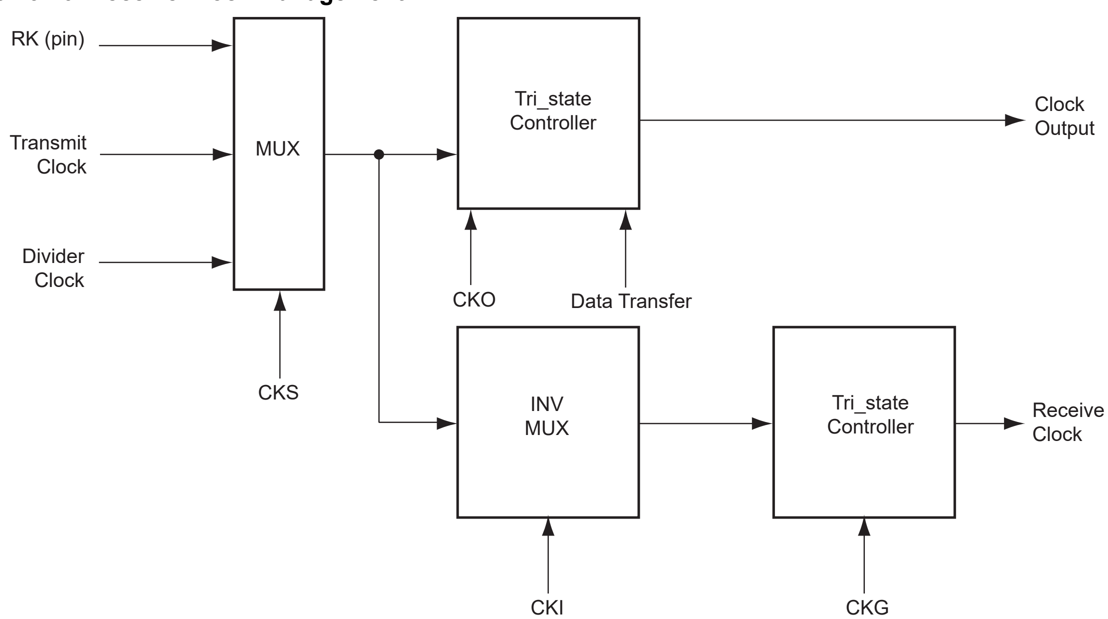


46.8.1.3 Receive Clock Management

The receive clock is generated from the transmit clock or the divider clock or an external clock scanned on the RK I/O pad. The Receive Clock is selected by the CKS field in SSC_RCMR (Receive Clock Mode Register). Receive Clocks can be inverted independently by the CKI bits in SSC_RCMR.

The receiver can also drive the RK I/O pad continuously or be limited to the current data transfer. The clock output is configured by the SSC_RCMR. The Receive Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_RCMR to select RK pin (CKS field) and at the same time Continuous Receive Clock (CKO field) can lead to unpredictable results.

Figure 46-10. Receive Clock Management



46.8.1.4 Serial Clock Ratio Considerations

The transmitter and the receiver can be programmed to operate with the clock signals provided on either the TK or RK pins. This allows the SSC to support many Client mode data transfers. In this case, the maximum clock speed allowed on the RK pin is:

- Peripheral clock divided by 2 if Receive Frame Synchronization is input
- Peripheral clock divided by 3 if Receive Frame Synchronization is output

In addition, the maximum clock speed allowed on the TK pin is:

- Peripheral clock divided by 7 if Transmit Frame Synchronization is input
- Peripheral clock divided by 2 if Transmit Frame Synchronization is output

These are only theoretical speed limits for first order calculations. Refer to the section "Electrical Characteristics" for exact speed limits on TK and RK.

46.8.2 Transmit Operations

A transmit frame is triggered by a start event and can be followed by synchronization data before data transmission.

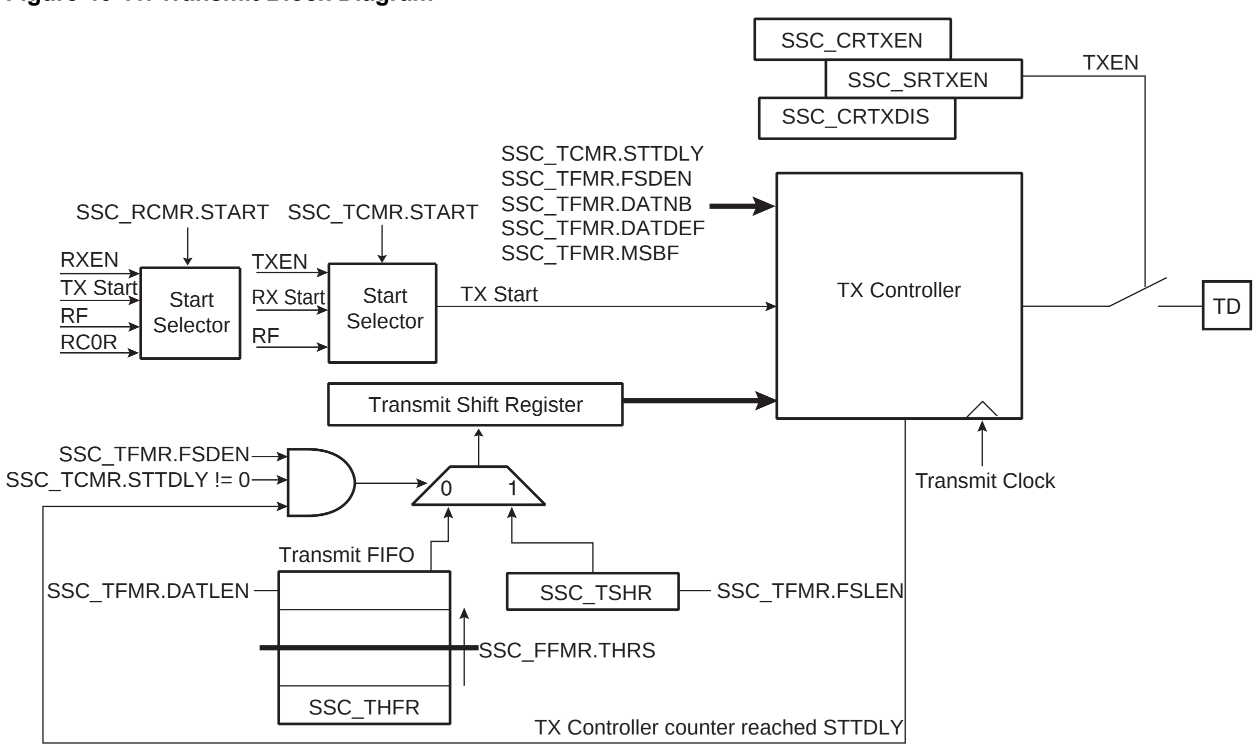
The start event is configured by setting the SSC_TCMR. See Start.

The frame synchronization is configured setting the Transmit Frame Mode Register (SSC_TFMR). See Frame Synchronization.

To transmit data, the transmitter uses a shift register clocked by the transmit clock signal and the start mode selected in the SSC_TCMR. Data are input by the application into the transmit FIFO by writing to the Transmit Holding FIFO register (SSC_THFR). Once the transmi t FIFO threshold programmed in the FIFO Mode register (SSC_FFMR.THRS) has been reached, data can be transferred to the transmit shift register according to the data format selected.

When both the transmit Holding FIFO and the transmit shift register are empty, the status flag TXEMPTY is set in the Status register (SSC_SR). As long as there is room left for writing new data into the transmit FIFO, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the Transmit Holding FIFO register. The number of remaining free data slots in the transmit FIFO is indicated by the SSC_SR.TXFRECNT field.

Figure 46-11. Transmit Block Diagram



46.8.3 Receive Operations

A receive frame is triggered by a start event and can be followed by synchronization data before data transmission.

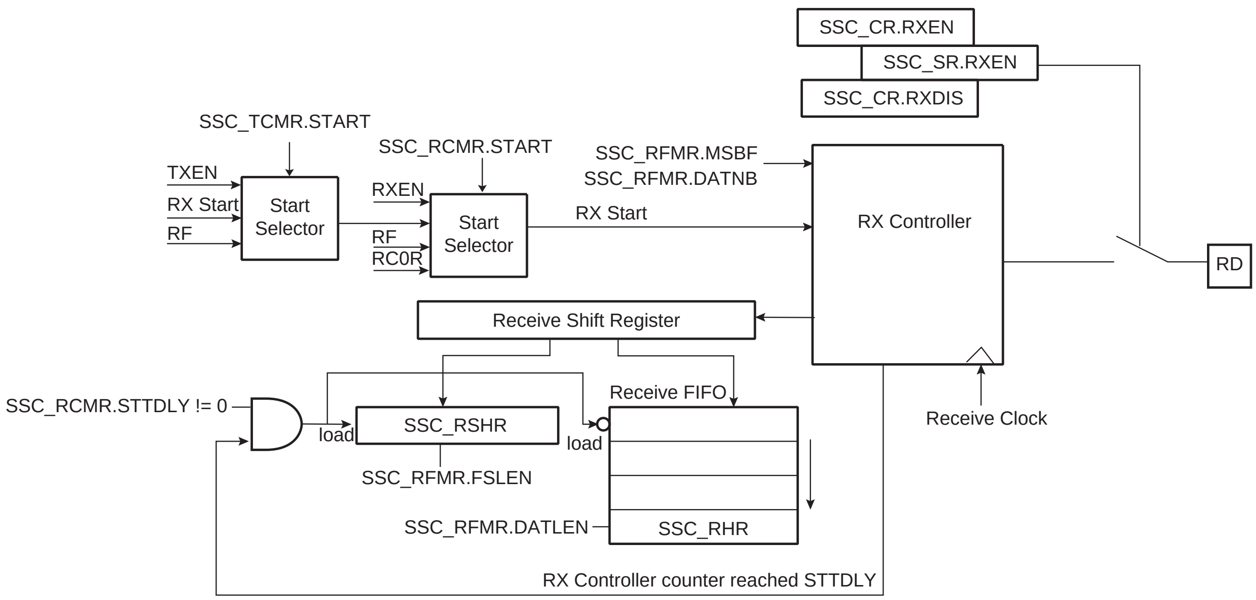
The start event is configured by setting the Receive Clock Mode Register (SSC_RCMR). See Start.

The frame synchronization is configured by setting the Receive Frame Mode Register (SSC_RFMR). See Frame Synchronization.

The receiver uses a shift register clocked by the receive clock signal and the start mode selected in the SSC_RCMR. The data is transferred from the shift register depending on the data format selected.

When the receive shift register is full, the SSC transfers the data into the receive FIFO, the status flag RXRDY is set in the SSC_SR and the data can be read in the Receive Holding FIFO register (SSC_RHFR). The number of remaining data words still to be read in the receive FIFO is indicated by the SSC_SR.RXURWCNT field. If a transfer occurs while the receive FIFO is full, the status flag OVRUN is set in the SSC_SR and the receive shift register is transferred in the receive FIFO. The oldest unread data is then lost.

Figure 46-12. Receive Block Diagram



46.8.4 Start

The transmitter and receiver can both be programmed to start their operations when an event occurs, respectively in the Transmit Start Selection (START) field of SSC_TCMR and in the Receive Start Selection (START) field of SSC_RCMR.

Under the following conditions the start event is independently programmable:

- Continuous. In this case, the transmission starts as soon as SSC_FFMR.THRS +1 words are written in SSC_THFR and the reception starts as soon as the receiver is enabled.
- Synchronously with the transmitter/receiver
- On detection of a falling/rising edge on TF/RF
- On detection of a low level/high level on TF/RF
- On detection of a level change or an edge on TF/RF

A start can be programmed in the same manner on either side of the Transmit/Receive Clock Register (SSC_RCMR/SSC_TCMR). Thus, the start could be on TF (Transmit) or RF (Receive).

Moreover, the receiver can start when data is detected in the bit stream with the Compare Functions.

Detection on TF/RF input/output is done by the field FSOS of the Transmit/Receive Frame Mode Register (SSC_TFMR/SSC_RFMR).

Figure 46-13. Transmit Start Mode

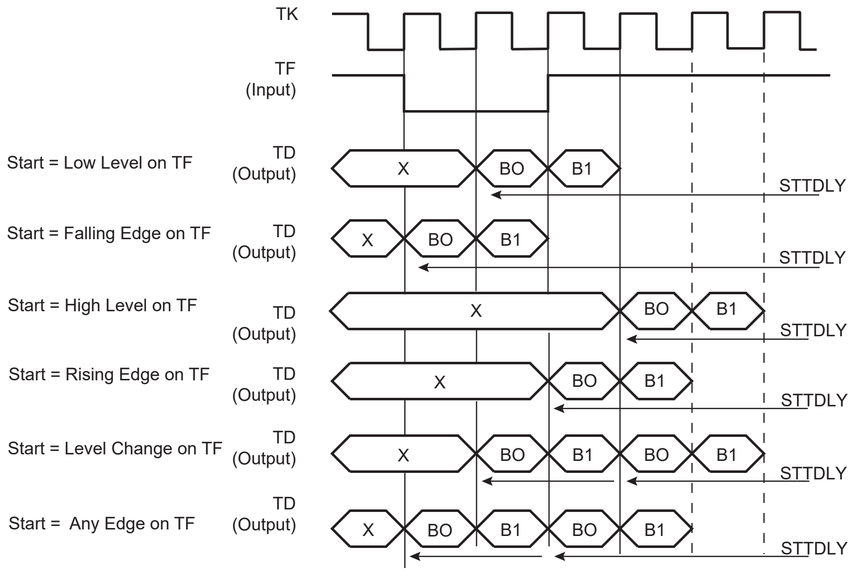
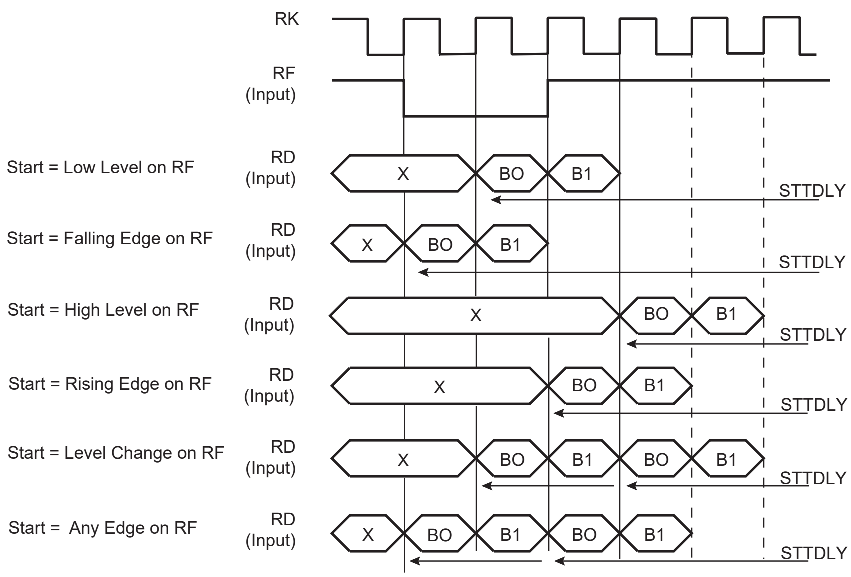


Figure 46-14. Receive Pulse/Edge Start Modes



46.8.5 Frame Synchronization

The Transmit and Receive Frame Sync pins, TF and RF, can be programmed to generate different kinds of Frame Sync signals. The Frame Sync Output Selection (FSOS) field in the Receive Frame Mode Register (SSC_RFMR) and in the Transmit Frame Mode Register (SSC_TFMR) are used to select the required waveform.

- Programmable low or high levels during data transfer are supported.
- Programmable high levels before the start of data transfers or toggling are also supported.

If a pulse waveform is selected, the Frame Sync Length (FSLEN) field in SSC_RFMR and SSC_TFMR programs the length of the pulse, from 1 bit time up to 256 bit times.

The periodicity of the Receive and Transmit Frame Sync pulse output can be programmed through the Period Divider Selection (PERIOD) field in SSC_RCMR and SSC_TCMR.

46.8.5.1 Frame Sync Data

Frame Sync Data transmits or receives a specific tag during the Frame Sync signal.

During the Frame Sync signal, the receiver can sample the RD line and store the data in the Receive Sync Holding Register and the transmitter can transfer Transmit Sync Holding Register in the shift register. The data length to be sampled/shifted out during the Frame Sync signal is programmed by the FSLEN field in SSC_RFMR/SSC_TFMR and has a maximum value of 256.

Concerning the Receive Frame Sync Data operation, if the Frame Sync Length is equal to or lower than the delay between the start event and the current data reception, the data sampling operation is performed in the Receive Sync Holding Register through the receive shift register.

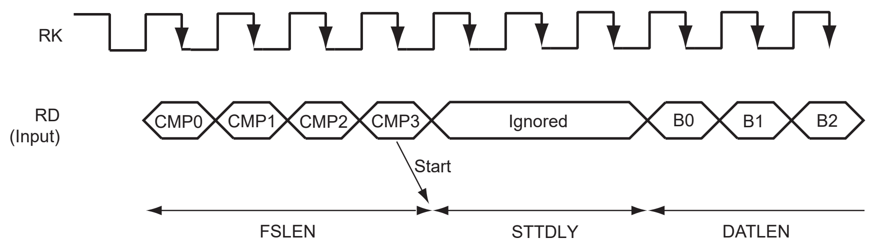
The Transmit Frame Sync Operation is performed by the transmitter only if the bit Frame Sync Data Enable (FSDEN) in SSC_TFMR is set. If the Frame Sync length is equal to or lower than the delay between the start event and the current data transmission, the normal transmission has priority and the data contained in the Transmit Sync Holding Register is transferred in the Transmit Register, then shifted out.

46.8.5.2 Frame Sync Edge Detection

The Frame Sync Edge detection is programmed by the FSEDGE field in SSC_RFMR/SSC_TFMR. This sets the corresponding flags RXSYN/TXSYN in the SSC Status Register (SSC_SR) on Frame Sync Edge detection (signals RF/TF).

46.8.6 Receive Compare Modes

Figure 46-15. Receive Compare Modes



46.8.6.1 Compare Functions

The length of the comparison patterns (Compare 0, Compare 1) and thus the number of bits they are compared to is defined by FSLEN, but with a maximum value of 256 bits. Comparison is always done by comparing the last bits received with the comparison pattern. Compare 0 can be one start event of the receiver. In this case, the receiver compares at each new sample the last bits received at the Compare 0 pattern contained in the Compare 0 Register (SSC_RC0R). When this start event is selected, the user can program the receiver to start a new data transfer either by writing a new Compare 0, or by receiving continuously until Compare 1 occurs. This selection is done with the STOP bit in the SSC_RCMR.

46.8.7 Data Format

The data framing format of both the transmitter and the receiver are programmable through the Transmitter Frame Mode Register (SSC_TFMR) and the Receive Frame Mode Register (SSC_RFMR). In either case, the user can independently select the following parameters:

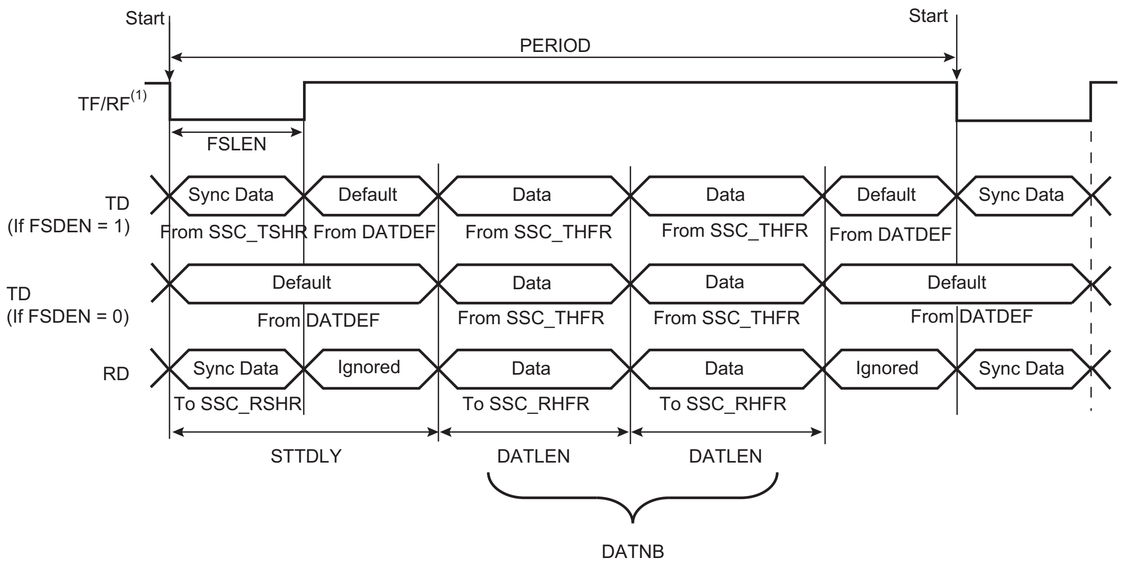
- Event that starts the data transfer (START)
- Delay in number of bit periods between the start event and the first data bit (STTDLY)
- Length of the data (DATLEN)
- Number of data to be transferred for each start event (DATNB)
- Length of synchronization transferred for each start event (FSLEN)
- Bit sense: most or least significant bit first (MSBF)

Additionally, the transmitter can be used to transfer synchronization and select the level driven on the TD pin while not in data transfer operation. This is done respectively by the Frame Sync Data Enable (FSDEN) and by the Data Default Value (DATDEF) bits in SSC_TFMR.

Table 46-2. Data Frame Registers

Transmitter	Receiver	Field	Length	Comment
SSC_TFMR	SSC_RFMR	DATLEN	Up to 32	Size of word
SSC_TFMR	SSC_RFMR	DATNB	Up to 16	Number of words transmitted in frame
SSC_TFMR	SSC_RFMR	MSBF	–	Most significant bit first
SSC_TFMR	SSC_RFMR	FSLEN	Up to 256	Size of Synchro data register
SSC_TFMR	–	DATDEF	0 or 1	Data default value ended
SSC_TFMR	–	FSDEN	–	Enable send SSC_TSHR
SSC_TCMR	SSC_RCMR	PERIOD	Up to 512	Frame size
SSC_TCMR	SSC_RCMR	STTDLY	Up to 255	Size of transmit start delay

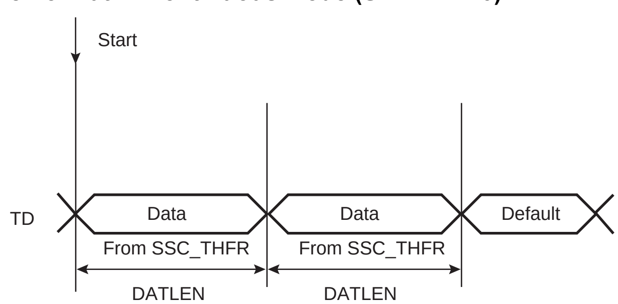
Figure 46-16. Transmit and Receive Frame Format in Edge/Pulse Start Modes



Note: 1. Example of input on falling edge of TF/RF.

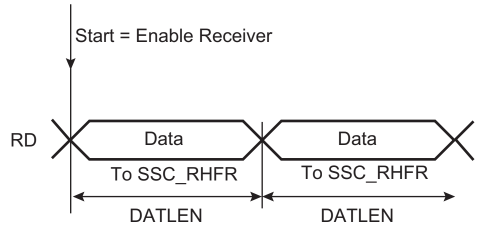
In the example illustrated above, the SSC_THFR is loaded twice. The FSDEN value has no effect on the transmission. SyncData cannot be output in Continuous mode.

Figure 46-17. Transmit Frame Format in Continuous Mode (STTDLY = 0)



Start: 1. TXEMPTY set to 1
2. Write into the SSC_THFR

Figure 46-18. Receive Frame Format in Continuous Mode (STTDLY = 0)



46.8.8 Loop Mode

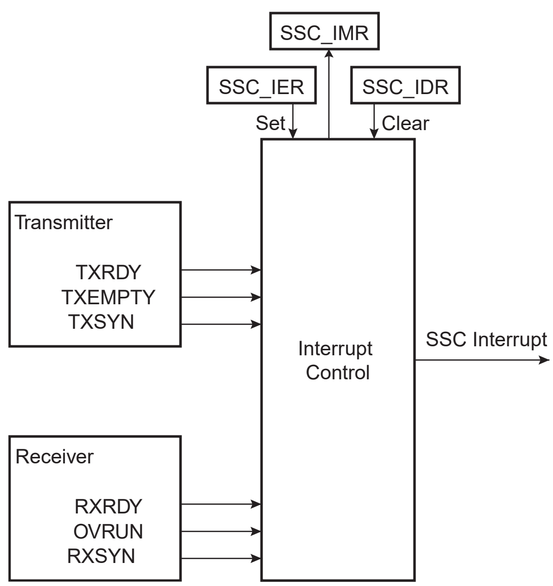
The receiver can be programmed to receive transmissions from the transmitter. This is done by setting the Loop Mode (LOOP) bit in the SSC_RFMR. In this case, RD is connected to TD, RF is connected to TF and RK is connected to TK.

46.8.9 Interrupt

Most bits in the SSC_SR have a corresponding bit in interrupt management registers.

The SSC can be programmed to generate an interrupt when it detects an event. The interrupt is controlled by writing the Interrupt Enable Register (SSC_IER) and Interrupt Disable Register (SSC_IDR). These registers enable and disable, respectively, the corresponding interrupt by setting and clearing the corresponding bit in the Interrupt Mask Register (SSC_IMR), which controls the generation of interrupts by asserting the SSC interrupt line connected to the interrupt controller.

Figure 46-19. Interrupt Block Diagram



46.8.10 Register Write Protection

To prevent any single software error from corrupting SSC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SSC Write Protection Mode Register (SSC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SSC Write Protection Status Register (SSC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SSC_WPSR.

The following registers can be write-protected:

- SSC Clock Mode Register
- SSC Receive Clock Mode Register
- SSC Receive Frame Mode Register

- SSC Transmit Clock Mode Register
- SSC Transmit Frame Mode Register
- SSC Receive Compare 0 Register
- SSC Receive Compare 1 Register

46.9 Register Summary

Note: Offsets 0x100–0x128 are reserved for PDC registers.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SSC_CR	31:24								
		23:16								
		15:8	SWRST						TXDIS	TXEN
		7:0							RXDIS	RXEN
0x04	SSC_CMR	31:24								
		23:16								
		15:8					DIV[11:8]			
		7:0	DIV[7:0]							
0x08 ... 0x0F	Reserved									
0x10	SSC_RCMR	31:24	PERIOD[7:0]							
		23:16	STTDLY[7:0]							
		15:8				STOP	START[3:0]			
		7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
0x14	SSC_RFMR	31:24	FSLEN_EXT[3:0]							FSEDGE
		23:16		FSOS[2:0]			FSLEN[3:0]			
		15:8					DATNB[3:0]			
		7:0	MSBF		LOOP	DATLEN[4:0]				
0x18	SSC_TCMR	31:24	PERIOD[7:0]							
		23:16	STTDLY[7:0]							
		15:8					START[3:0]			
		7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
0x1C	SSC_TFMR	31:24	FSLEN_EXT[3:0]							FSEDGE
		23:16	FSDEN	FSOS[2:0]			FSLEN[3:0]			
		15:8					DATNB[3:0]			
		7:0	MSBF		DATDEF	DATLEN[4:0]				
0x20	SSC_RHFR	31:24	RDAT[31:24]							
		23:16	RDAT[23:16]							
		15:8	RDAT[15:8]							
		7:0	RDAT[7:0]							
0x24	SSC_THFR	31:24	TDAT[31:24]							
		23:16	TDAT[23:16]							
		15:8	TDAT[15:8]							
		7:0	TDAT[7:0]							
0x28	SSC_FFMR	31:24								
		23:16								RXFIFODIS
		15:8					THRS[3:0]			
		7:0								TXFIFODIS
0x2C ... 0x2F	Reserved									
0x30	SSC_RSHR	31:24								
		23:16								
		15:8	RSDAT[15:8]							
		7:0	RSDAT[7:0]							
0x34	SSC_TSHR	31:24								
		23:16								
		15:8	TSDAT[15:8]							
		7:0	TSDAT[7:0]							
0x38	SSC_RC0R	31:24								
		23:16								
		15:8	CP0[15:8]							
		7:0	CP0[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	SSC_RC1R	31:24								
		23:16								
		15:8	CP1[15:8]							
		7:0	CP1[7:0]							
0x40	SSC_SR	31:24	RXURWCNT[3:0]				TXFRECNT[3:0]			
		23:16							RXEN	TXEN
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x44	SSC_IER	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x48	SSC_IDR	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x4C	SSC_IMR	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x50 ... 0xE3	Reserved									
0xE4	SSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SSC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

46.9.1 SSC Control Register

Name: SSC_CR
Offset: 0x0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWRST						TXDIS	TXEN
Access	W						W	W
Reset	–						–	–

Bit	7	6	5	4	3	2	1	0
							RXDIS	RXEN
Access							W	W
Reset							–	–

Bit 15 – SWRST Software Reset

Value	Description
0	No effect.
1	Performs a software reset. Has priority on any other bit in SSC_CR.

Bit 9 – TXDIS Transmit Disable

Value	Description
0	No effect.
1	Disables Transmit. If a character is currently being transmitted, disables at end of current character transmission.

Bit 8 – TXEN Transmit Enable

Value	Description
0	No effect.
1	Enables Transmit if TXDIS is not set.

Bit 1 – RXDIS Receive Disable

Value	Description
0	No effect.
1	Disables Receive. If a character is currently being received, disables at end of current character reception.

Bit 0 – RXEN Receive Enable

Value	Description
0	No effect.
1	Enables Receive if RXDIS is not set.

46.9.2 SSC Clock Mode Register

Name: SSC_CMR
Offset: 0x4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					DIV[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – DIV[11:0] Clock Divider

Value	Description
0	The Clock Divider is not active.
Any other value	The divided clock equals the peripheral clock divided by 2 times DIV. The maximum bit rate is $f_{peripheral\ clock}/2$. The minimum bit rate is $f_{peripheral\ clock}/2 \times 4095 = f_{peripheral\ clock}/8190$.

46.9.3 SSC Receive Clock Mode Register

Name: SSC_RCMR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	STTDLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				STOP	START[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PERIOD[7:0] Receive Period Divider Selection
This field selects the divider to apply to the selected Receive Clock in order to generate a new Frame Sync signal. If 0, no PERIOD signal is generated. If not 0, a PERIOD signal is generated each 2 x (PERIOD + 1) Receive Clock.

Bits 23:16 – STTDLY[7:0] Receive Start Delay
If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the current start of reception. When the receiver is programmed to start synchronously with the transmitter, the delay is also applied.
Note: STTDLY must be configured in relation to the receive synchronization data to be stored in SSC_RSHR.

Bit 12 – STOP Receive Stop Selection

Value	Description
0	After completion of a data transfer when starting with a Compare 0, the receiver stops the data transfer and waits for a new compare 0.
1	After starting a receive with a Compare 0, the receiver operates in a continuous mode until a Compare 1 is detected.

Bits 11:8 – START[3:0] Receive Start Selection

Value	Name	Description
0	CONTINUOUS	Continuous, as soon as the receiver is enabled, and immediately after the end of transfer of the previous data.
1	TRANSMIT	Transmit start
2	RF_LOW	Detection of a low level on RF signal
3	RF_HIGH	Detection of a high level on RF signal
4	RF_FALLING	Detection of a falling edge on RF signal
5	RF_RISING	Detection of a rising edge on RF signal
6	RF_LEVEL	Detection of any level change on RF signal
7	RF_EDGE	Detection of any edge on RF signal
8	CMP_0	Compare 0

Bits 7:6 – CKG[1:0] Receive Clock Gating Selection

Value	Name	Description
0	CONTINUOUS	None
1	EN_RF_LOW	Receive Clock enabled only if RF Low
2	EN_RF_HIGH	Receive Clock enabled only if RF High

Bit 5 – CKI Receive Clock Inversion

CKI affects only the Receive Clock and not the output clock signal.

Value	Description
0	The data inputs (Data and Frame Sync signals) are sampled on Receive Clock falling edge. The Frame Sync signal output is shifted out on Receive Clock rising edge.
1	The data inputs (Data and Frame Sync signals) are sampled on Receive Clock rising edge. The Frame Sync signal output is shifted out on Receive Clock falling edge.

Bits 4:2 – CKO[2:0] Receive Clock Output Mode Selection

Value	Name	Description
0	NONE	None, RK pin is an input
1	CONTINUOUS	Continuous Receive Clock, RK pin is an output
2	TRANSFER	Receive Clock only during data transfers, RK pin is an output

Bits 1:0 – CKS[1:0] Receive Clock Selection

Value	Name	Description
0	MCK	Divided Clock
1	TK	TK Clock signal
2	RK	RK pin

46.9.4 SSC Receive Frame Mode Register

Name: SSC_RFMR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FSLEN_EXT[3:0]							FSEDGE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
		FSOS[2:0]			FSLEN[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DATNB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBF		LOOP	DATLEN[4:0]				
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:28 – FSLEN_EXT[3:0] FSLEN Field Extension
Extends FSLEN field. For details, see FSLEN: Receive Frame Sync Length.

Bit 24 – FSEDGE Frame Sync Edge Detection
Determines which edge on Frame Sync will generate the interrupt RXSYN in the SSC Status Register.

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

Bits 22:20 – FSOS[2:0] Receive Frame Sync Output Selection

Value	Name	Description
0	NONE	None, RF pin is an input
1	NEGATIVE	Negative Pulse, RF pin is an output
2	POSITIVE	Positive Pulse, RF pin is an output
3	LOW	Driven Low during data transfer, RF pin is an output
4	HIGH	Driven High during data transfer, RF pin is an output
5	TOGGLING	Toggling at each start of data transfer, RF pin is an output

Bits 19:16 – FSLEN[3:0] Receive Frame Sync Length
This field defines the number of bits sampled and stored in the Receive Sync Data Register. When this mode is selected by the START field in the Receive Clock Mode Register, it also determines the length of the sampled data to be compared to the Compare 0 or Compare 1 register.
This field is used with FSLEN_EXT to determine the pulse length of the Receive Frame Sync signal.
Pulse length is equal to FSLEN + (FSLEN_EXT × 16) + 1 Receive Clock periods.

Bits 11:8 – DATNB[3:0] Data Number per Frame
This field defines the number of data words to be received after each transfer start, which is equal to (DATNB + 1).

Bit 7 – MSBF Most Significant Bit First

Value	Description
0	The lowest significant bit of the data register is sampled first in the bit stream.
1	The most significant bit of the data register is sampled first in the bit stream.

Bit 5 – LOOP Loop Mode

Value	Description
0	Normal operating mode.
1	RD is driven by TD, RF is driven by TF and TK drives RK.

Bits 4:0 – DATLEN[4:0] Data Length

Value	Description
0	Forbidden value (1-bit data length not supported).
Any other value	The bit stream contains DATLEN + 1 data bits.

46.9.5 SSC Transmit Clock Mode Register

Name: SSC_TCMR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	STTDLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					START[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PERIOD[7:0] Transmit Period Divider Selection
This field selects the divider to apply to the selected Transmit Clock to generate a new Frame Sync signal. If 0, no period signal is generated. If not 0, a period signal is generated at each 2 × (PERIOD + 1) Transmit Clock.

Bits 23:16 – STTDLY[7:0] Transmit Start Delay
If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the current start of transmission of data. When the transmitter is programmed to start synchronously with the receiver, the delay is also applied.
Note:
If STTDLY is too short with respect to transmit synchronization data (SSC_TSHR), SSC_THR.TDAT is transmitted instead of the end of SSC_TSHR.

Bits 11:8 – START[3:0] Transmit Start Selection

Value	Name	Description
0	CONTINUOUS	Continuous, as soon as a word is written in the SSC_THR (if Transmit is enabled), and immediately after the end of transfer of the previous data
1	RECEIVE	Receive start
2	TF_LOW	Detection of a low level on TF signal
3	TF_HIGH	Detection of a high level on TF signal
4	TF_FALLING	Detection of a falling edge on TF signal
5	TF_RISING	Detection of a rising edge on TF signal
6	TF_LEVEL	Detection of any level change on TF signal
7	TF_EDGE	Detection of any edge on TF signal

Bits 7:6 – CKG[1:0] Transmit Clock Gating Selection

Value	Name	Description
0	CONTINUOUS	None
1	EN_TF_LOW	Transmit Clock enabled only if TF Low

Value	Name	Description
2	EN_TF_HIGH	Transmit Clock enabled only if TF High

Bit 5 – CKI Transmit Clock Inversion
CKI affects only the Transmit Clock and not the Output Clock signal.

Value	Description
0	The data outputs (Data and Frame Sync signals) are shifted out on Transmit Clock falling edge. The Frame Sync signal input is sampled on Transmit Clock rising edge.
1	The data outputs (Data and Frame Sync signals) are shifted out on Transmit Clock rising edge. The Frame Sync signal input is sampled on Transmit Clock falling edge.

Bits 4:2 – CKO[2:0] Transmit Clock Output Mode Selection

Value	Name	Description
0	NONE	None, TK pin is an input
1	CONTINUOUS	Continuous Transmit Clock, TK pin is an output
2	TRANSFER	Transmit Clock only during data transfers, TK pin is an output

Bits 1:0 – CKS[1:0] Transmit Clock Selection

Value	Name	Description
0	MCK	Divided Clock
1	RK	RK Clock signal
2	TK	TK pin

46.9.6 SSC Transmit Frame Mode Register

Name: SSC_TFMR
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FSLEN_EXT[3:0]							FSEDGE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
	FSDEN	FSOS[2:0]			FSLEN[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DATNB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBF		DATDEF	DATLEN[4:0]				
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:28 – FSLEN_EXT[3:0] FSLEN Field Extension
Extends FSLEN field. For details, see FSLEN description below.

Bit 24 – FSEDGE Frame Sync Edge Detection
Determines which edge on frame synchronization will generate the interrupt TXSYN (Status Register).

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

Bit 23 – FSDEN Frame Sync Data Enable

Value	Description
0	The TD line is driven with the default value during the Transmit Frame Sync signal.
1	SSC_TSHR value is shifted out during the transmission of the Transmit Frame Sync signal.

Bits 22:20 – FSOS[2:0] Transmit Frame Sync Output Selection

Value	Name	Description
0	NONE	None, TF pin is an input
1	NEGATIVE	Negative Pulse, TF pin is an output
2	POSITIVE	Positive Pulse, TF pin is an output
3	LOW	Driven Low during data transfer
4	HIGH	Driven High during data transfer
5	TOGGLING	Toggling at each start of data transfer

Bits 19:16 – FSLEN[3:0] Transmit Frame Sync Length
This field defines the length of the Transmit Frame Sync signal and the number of bits shifted out from SSC_TSHR if FSDEN is 1.
This field is used with FSLEN_EXT to determine the pulse length of the Transmit Frame Sync signal.
Pulse length is equal to FSLEN + (FSLEN_EXT × 16) + 1 Transmit Clock period.

Bits 11:8 – DATNB[3:0] Data Number per Frame
This field defines the number of data words to be transferred after each transfer start, which is equal to (DATNB + 1).

Bit 7 – MSBF Most Significant Bit First

Value	Description
0	The lowest significant bit of the data register is shifted out first in the bit stream.
1	The most significant bit of the data register is shifted out first in the bit stream.

Bit 5 – DATDEF Data Default Value
This bit defines the level driven on the TD pin while out of transmission. Note that if the pin is defined as multi-drive by the PIO Controller, the pin is enabled only if the SCC TD output is 1.
When the TD pin is configured in Multi-drive (Open-drain) mode by the PIO controller, a 0 is driven if SSC data output equals 0 and the pin is in high-impedance when SSC data output is 1.

Bits 4:0 – DATLEN[4:0] Data Length

Value	Description
0	Forbidden value (1-bit data length not supported).
Any other value	The bit stream contains DATLEN + 1 data bits.

46.9.7 SSC Receive Holding FIFO Register

Name: SSC_RHFR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RDAT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RDAT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RDAT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDAT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RDAT[31:0] Receive Data
Right-aligned regardless of the number of data bits defined by SSC_RFMR.DATLEN.

46.9.8 SSC Transmit Holding FIFO Register

Name: SSC_THFR
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	TDAT[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TDAT[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TDAT[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TDAT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – TDAT[31:0] Transmit Data
Right-aligned regardless of the number of data bits defined by SSC_TFMR.DATLEN.

46.9.9 SSC FIFO Mode Register

Name: SSC_FFMR
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								RXFIFODIS
Access								R/W
Reset								0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
					THRS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								TXFIFODIS
Access								R/W
Reset								0

Bit 16 – RXFIFODIS Receive FIFO Disable

Value	Description
0	The receive FIFO is enabled.
1	The receive FIFO is disabled. Only a single entry Receive Holding register is available instead.

Bits 11:8 – THRS[3:0] Transmit Start Threshold
Number of additional data to be written into SCC_THFR prior to loading the first data into the Transmit Shift register.
0 means the first data written into SCC_THFR is immediately loaded into the Shift register.
7 means the loading into SSC_THFR will occur only when the transmit FIFO is full.

Bit 0 – TXFIFODIS Transmit FIFO Disable

Value	Description
0	The transmit FIFO is enabled.
1	The transmit FIFO is disabled. Only a single entry Transmit Holding register is available instead.

46.9.10 SSC Receive Synchronization Holding Register

Name: SSC_RSHR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSDAT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RSDAT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RSDAT[15:0] Receive Synchronization Data

46.9.11 SSC Transmit Synchronization Holding Register

Name: SSC_TSHR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TSDAT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSDAT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TSDAT[15:0] Transmit Synchronization Data

46.9.12 SSC Receive Compare 0 Register

Name: SSC_RC0R
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CP0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CP0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CP0[15:0] Receive Compare Data 0

46.9.13 SSC Receive Compare 1 Register

Name: SSC_RC1R
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CP1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CP1[15:0] Receive Compare Data 1

46.9.14 SSC Status Register

Name: SSC_SR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXURWCNT[3:0]				TXFRECNT[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			R	R			R	R
Reset			0	0			0	0

Bits 31:28 – RXURWCNT[3:0] Receive FIFO Unread Word Count
Number of received data words in the receive FIFO still to be read from SSC_RHFR.
This counter is incremented upon each load of the receive FIFO with a data from the receive shifter and decremented by each data read from SSC_RHFR.

Bits 27:24 – TXFRECNT[3:0] Transmit FIFO Free Entries Count
Number of free data entries in the transmit FIFO that can be written into SSC_THFR.
This counter is decremented by each data write into SSC_THFR and incremented upon each transfer of a transmit FIFO data into the transmit shifter.

Bit 17 – RXEN Receive Enable

Value	Description
0	Receive is disabled.
1	Receive is enabled.

Bit 16 – TXEN Transmit Enable

Value	Description
0	Transmit is disabled.
1	Transmit is enabled.

Bit 11 – RXSYN Receive Sync

Value	Description
0	An Rx Sync has not occurred since the last read of the Status register.
1	An Rx Sync has occurred since the last read of the Status register.

Bit 10 – TXSYN Transmit Sync

Value	Description
0	A Tx Sync has not occurred since the last read of the Status register.
1	A Tx Sync has occurred since the last read of the Status register.

Bit 9 – CP1 Compare 1

Value	Description
0	A compare 1 has not occurred since the last read of the Status register.
1	A compare 1 has occurred since the last read of the Status register.

Bit 8 – CP0 Compare 0

Value	Description
0	A compare 0 has not occurred since the last read of the Status register.
1	A compare 0 has occurred since the last read of the Status register.

Bit 5 – OVRUN Receive Overrun

Value	Description
0	No data has been loaded in SSC_RHR while previous data has not been read since the last read of the Status Register.
1	Data has been loaded in SSC_RHR while previous data has not yet been read since the last read of the Status Register.

Bit 4 – RXRDY Receive Ready

Value	Description
0	SSC_RHR is empty.
1	Data has been received and loaded in SSC_RHR.

Bit 1 – TXEMPTY Transmit Empty

Value	Description
0	Data remains in SSC_THR or is currently transmitted from TSR.
1	Last data written in SSC_THR has been loaded in TSR and last data loaded in TSR has been transmitted.

Bit 0 – TXRDY Transmit Ready

Value	Description
0	Data has been loaded in SSC_THR and is waiting to be loaded in the Transmit Shift register (TSR).
1	SSC_THR is empty.

46.9.15 SSC Interrupt Enable Register

Name: SSC_IER
Offset: 0x44
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit 11 – RXSYN Rx Sync Interrupt Enable

Value	Description
0	No effect.
1	Enables the Rx Sync Interrupt.

Bit 10 – TXSYN Tx Sync Interrupt Enable

Value	Description
0	No effect.
1	Enables the Tx Sync Interrupt.

Bit 9 – CP1 Compare 1 Interrupt Enable

Value	Description
0	No effect.
1	Enables the Compare 1 Interrupt.

Bit 8 – CP0 Compare 0 Interrupt Enable

Value	Description
0	No effect.
1	Enables the Compare 0 Interrupt.

Bit 5 – OVRUN Receive Overrun Interrupt Enable

Value	Description
0	No effect.
1	Enables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Enable

Value	Description
0	No effect.

Value	Description
1	Enables the Receive Ready Interrupt.

Bit 1 – TXEMPTY Transmit Empty Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Empty Interrupt.

Bit 0 – TXRDY Transmit Ready Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Ready Interrupt.

46.9.16 SSC Interrupt Disable Register

Name: SSC_IDR
Offset: 0x48
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit 11 – RXSYN Rx Sync Interrupt Enable

Value	Description
0	No effect.
1	Disables the Rx Sync Interrupt.

Bit 10 – TXSYN Tx Sync Interrupt Enable

Value	Description
0	No effect.
1	Disables the Tx Sync Interrupt.

Bit 9 – CP1 Compare 1 Interrupt Disable

Value	Description
0	No effect.
1	Disables the Compare 1 Interrupt.

Bit 8 – CP0 Compare 0 Interrupt Disable

Value	Description
0	No effect.
1	Disables the Compare 0 Interrupt.

Bit 5 – OVRUN Receive Overrun Interrupt Disable

Value	Description
0	No effect.
1	Disables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Disable

Value	Description
0	No effect.

Value	Description
1	Disables the Receive Ready Interrupt.

Bit 1 – TXEMPTY Transmit Empty Interrupt Disable

Value	Description
0	No effect.
1	Disables the Transmit Empty Interrupt.

Bit 0 – TXRDY Transmit Ready Interrupt Disable

Value	Description
0	No effect.
1	Disables the Transmit Ready Interrupt.

46.9.17 SSC Interrupt Mask Register

Name: SSC_IMR
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			R	R			R	R
Reset			0	0			0	0

Bit 11 – RXSYN Rx Sync Interrupt Mask

Value	Description
0	The Rx Sync Interrupt is disabled.
1	The Rx Sync Interrupt is enabled.

Bit 10 – TXSYN Tx Sync Interrupt Mask

Value	Description
0	The Tx Sync Interrupt is disabled.
1	The Tx Sync Interrupt is enabled.

Bit 9 – CP1 Compare 1 Interrupt Mask

Value	Description
0	The Compare 1 Interrupt is disabled.
1	The Compare 1 Interrupt is enabled.

Bit 8 – CP0 Compare 0 Interrupt Mask

Value	Description
0	The Compare 0 Interrupt is disabled.
1	The Compare 0 Interrupt is enabled.

Bit 5 – OVRUN Receive Overrun Interrupt Mask

Value	Description
0	The Receive Overrun Interrupt is disabled.
1	The Receive Overrun Interrupt is enabled.

Bit 4 – RXRDY Receive Ready Interrupt Mask

Value	Description
0	The Receive Ready Interrupt is disabled.

Value	Description
1	The Receive Ready Interrupt is enabled.

Bit 1 – TXEMPTY Transmit Empty Interrupt Mask

Value	Description
0	The Transmit Empty Interrupt is disabled.
1	The Transmit Empty Interrupt is enabled.

Bit 0 – TXRDY Transmit Ready Interrupt Mask

Value	Description
0	The Transmit Ready Interrupt is disabled.
1	The Transmit Ready Interrupt is enabled.

46.9.18 SSC Write Protection Mode Register

Name: SSC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535343	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535343 (“SSC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x535343 (“SSC” in ASCII).

46.9.19 SSC Write Protection Status Register

Name: SSC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protect Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the SSC_WPSR.
1	A write protection violation has occurred since the last read of the SSC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

47. Sony/Philips Digital Interface Receiver (SPDIFRX)

47.1 Description

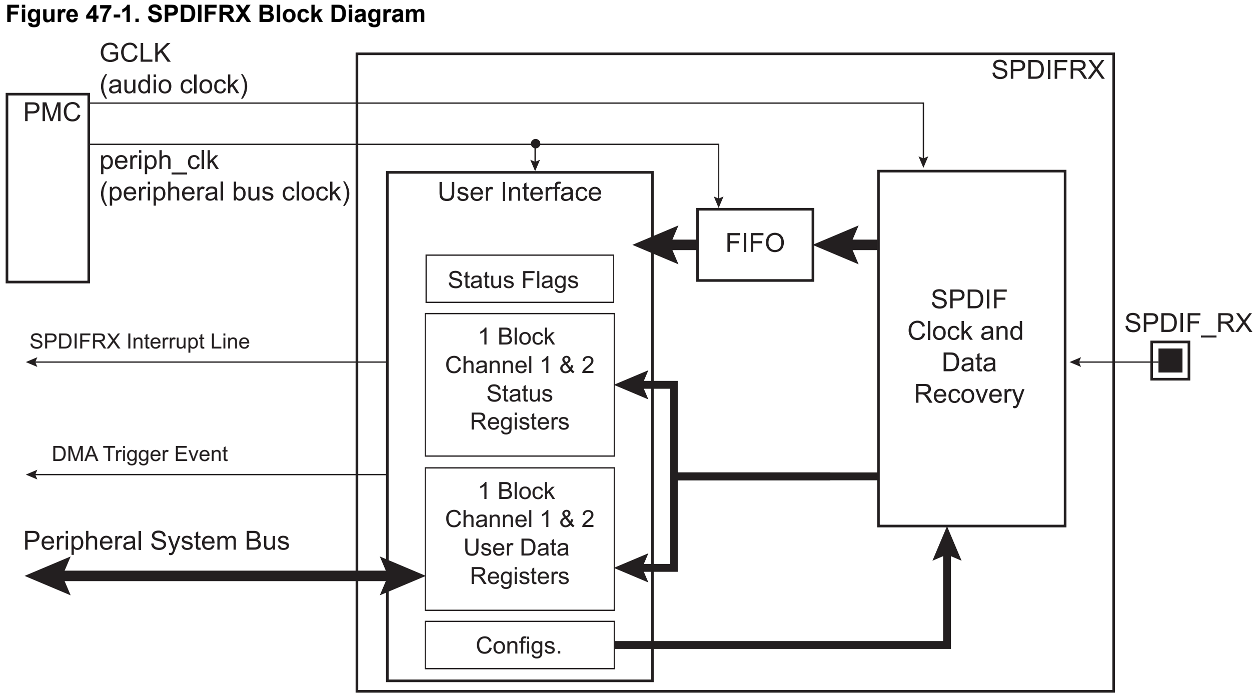
The Sony/Philips Digital Interface Receiver (SPDIFRX) is a serial port compliant with the IEC-60958 standard.

47.2 Embedded Characteristics

- SPDIF/AES-EBU Compatible Serial Port
- 32 Samples FIFO
- Data Width Configurable to 24 bits, 20 bits or 16 bits
- Packed and Unpacked Data Support for System Memory Optimization
- Line State Events Report and Source of Interrupt
- Line Error Rate Report
- Full Memory Map of 192 bits for Channel 1 and Channel 2 Status and User Data
- First 32-bit Status A, Status B Change Report and Source of Interrupt
- Line Digital Filter
- Register Write Protection
- Abnormal Software Access and Internal Sequencer Integrity Check Reports

47.3 Block Diagram

Figure 47-1. SPDIFRX Block Diagram



47.4 Signal Description

Table 47-1. Pin Description

Pin Name	Description	Direction
SPDIF_RX	SPDIF Receive Line	Input

47.5 Product Dependencies

47.5.1 I/O Lines

The pins used for interfacing the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the SPDIF pin to their peripheral functions.

47.5.2 Power Management

The SPDIFRX is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the SPDIFRX clocks.

47.5.3 Interrupt Sources

The SPDIFRX interface has an interrupt line connected to the interrupt controller. Handling the SPDIFRX interrupt requires programming the interrupt controller before configuring the SPDIFRX.

47.6 Functional Description

47.6.1 Protocol

The SPDIFRX protocol is defined in IEC-60958 standard.

Data is performed by the transfer of blocks of 192 frames.

Each frame is divided into two subframes, one for each channel.

Each subframe is divided into 32 time slots carrying the following fields:

- Preamble (4 bits)–Three possible preambles named B, M and W. Preambles are specific patterns providing synchronization and identification of the subframes and blocks.
- Data (24 bits)–Can carry a PCM audio data or non-PCM data streams.
- Validity bit
- User data bit
- Channel status bit
- Parity bit–Ensures that time slots 4 to 31 carry an even number of ones and an even number of zeros.

Note: The data field may carry 16-bit, 20-bit or 24-bit data.

Figure 47-2. SPDIFRX Frame and Block Format

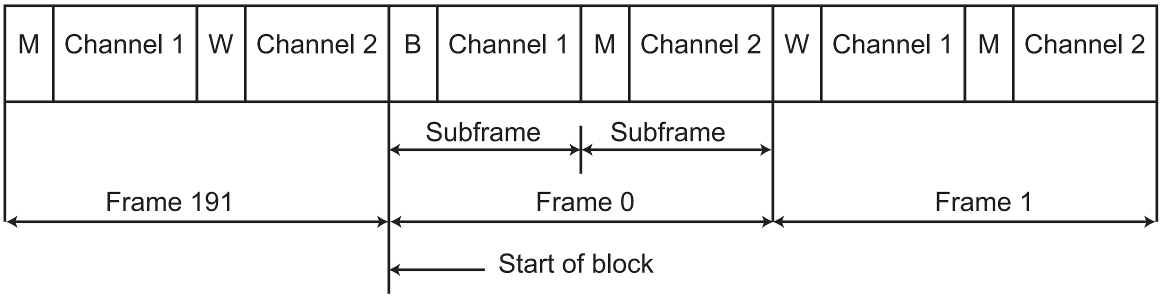
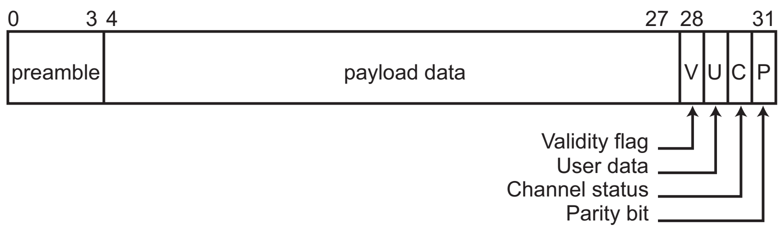


Figure 47-3. SPDIFRX Subframe Format



47.6.2 Data Rate

The SPDIFRX operates at the frequency of generic clock (GCLK) input clock (typically the audio clock). The GCLK clock frequency must be greater than 768 × f_S (12 × 64 × f_S). For example, 192 kHz sampling frequency can be recovered if the GCLK clock frequency is 150 MHz and above, and f_S = 48 kHz can be recovered if the GCLK clock frequency is above 38 MHz.

47.6.3 Clock and Data Recovery

When the SPDIFRX is first enabled by writing a '1' in the bit RXEN in the Mode register (SPDIFRX_MR), the bit ULOCK in the Receiver Status register (SPDIFRX_RSR) is set, indicating that the synchronization of the receiver circuitry with the received stream is in progress. The SPDIFRX_RSR.ULOCK=1 if no synchronization is possible.

If there is no signal (no detection of falling/rising edges) on the SPDIF_RX line, the bit SPDIFRX_RSR.NOSIGNAL='1'.

If the SPDIF_RX line carries an SPDIF signal having a sampling frequency which does not meet the criteria defined in 47.6.2. Data Rate, the bit SPDIFRX_RSR.LOWF= '1'.

If a non-SPDIF format is received/detected, the bit SPDIFRX_RSR.BADF= '1'. A non-SPDIF format is detected when the width of the larger pulse (located in preamble area) is not higher than 2.5 times the width of the minimum pulse or is greater than 4 times the width of the minimum pulse.

As soon as a valid SPDIF format is detected, the clock recovery circuitry is locked and the bi-mark data is decoded and the circuitry searches for predefined preamble values.

Once one of the preamble values is found, SPDIFRX_RSR.ULOCK is cleared and the bit LOCKED in the Interrupt Status register (SPDIFRX_ISR) is set to '1'. An interrupt can be triggered if LOCKED in the Interrupt Enable register (SPDIFRX_IER) is written to '1'.

The SPDIF_RX line is filtered by a digital filter to reduce noise before clock recovery. Any pulse with width lower than or equal to one GCLK clock period is eliminated.

The maximum lock time is 1.5 sample periods (1.5 frames). As soon as SPDIFRX_RSR.ULOCK is cleared, the payload data is stored in the FIFO and the channel 1 and 2 status bits and user data bits are stored. At the end of each block (192 frames), the bit SPDIFRX_ISR.BLOCKEND is set to '1'. If the SPDIFRX_ISR.BLOCKEND=1, the content of channel status (SPDIFRX_CH1SRx[x=0..5], SPDIFRX_CH2SRx[x=0..5]) and user data (SPDIFRX_CH1UDRx[x=0..5], SPDIFRX_CH2UDRx[x=0..5]) registers is valid.

When locked, in case of a loss of signal (32 consecutive 0's or 1's), the flag SPDIFRX_ISR.LOSS is set to '1' and the recovery circuitry searches again for a synchronization pattern. When unlocked (SPDIFRX_RSR.ULOCK=1) the bit SPDIFRX_RSR.NOSIGNAL is set to '1'.

Several error types are monitored and reported in the SPDIFRX_ISR:
- If the flag SPDIFRX_ISR.PAR_ERR is set to '1', the value of one of the parity bit does not match the parity of the corresponding data received.
- If the flag SPDIFRX_ISR.NRZ_ERR is set to '1', a bi-mark error has been detected.
- If the flag SPDIFRX_ISR.PRE_ERR is set to '1', a preamble error has been detected.

The data rate is monitored on-the-fly and any change is reported in the flag SPDIFRX_ISR.FSE.

If the data rate change does not create an important loss of synchronization, the clock recovery logic is reloaded with the new values (see 47.6.4. Sample Frequency Measurement), thus providing a fast data recovery time. Anyway, in this case, a preamble error or a parity error can be detected.

A major data rate change can incur the loss of the clock recovery circuitry, resulting in preamble errors. It 16 consecutive preamble errors are detected, the clock recovery circuitry is automatically restarted, the flag SPDIFRX_ISR.FSE is set to "1" and as soon as circuitry is synchronized again, SPDIFRX_ISR.LOCK='1'.

The potential causes of unlocked circuitry are provided in SPDIFRX_RSR when SPDIFRX_RSR.ULOCK is set to '1'. If there is no signal (no pulse detectable on the SPDIF RX line), SPDIFRX_RSR.NOSIGNAL is set to '1'. If the SPDIF_RX line carries a signal but the format cannot be analyzed as an SPDIFRX format, SPDIFRX_RSR.BADF is set to '1'. If the SPDIFRX format is correct but the GCLK frequency provided to the SPDIFRX module does not meet the minimum frequency conditions for the sample rate frequency carried on the SPDIF_RX line, SPDIFRX_RSR.LOWF is set to '1'.

47.6.4 Sample Frequency Measurement

The SPDIFRX performs a measurement of the width of the bi-mark code pulses and reports the value in SPDIFRX_RSR.IFS. When the receiver is synchronized (SPDIFRX_RSR.ULOCK=0), the value of the field IFS corresponds to the number of GCLK periods measured for a period of 2 bits (source coding). This value can be used to determine the sampling frequency carried on the SPDIF_RX line (see 47.7.7. SPDIFRX_RSR).

47.6.5 Data Management

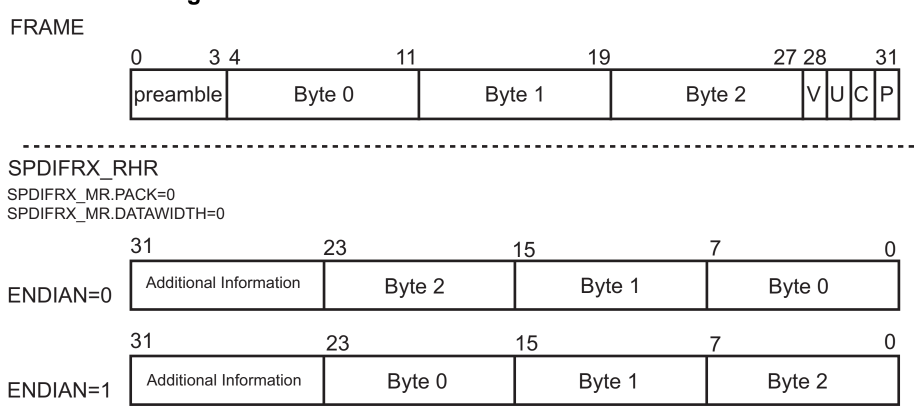
The payload data is stored for channels 1 and 2. By writing a '1' to SPDIFRX_MR.VBMODE, the receiver can be configured to automatically discard any sample with an associated validity bit set to '1'.

The endianness of the payload data can be modified during the store operation by configuring SPDIFRX_MR.ENDIAN.

Note: SPDIFRX_MR.ENDIAN applies only when SPDIFRX_MR.DATAWIDTH is set to 0 (24-bit mode).

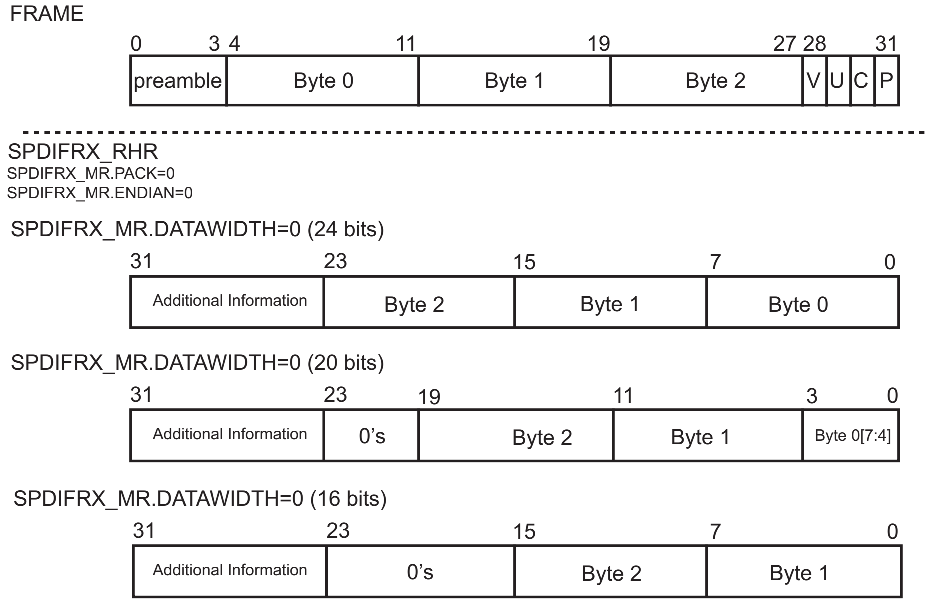
See the figure below.

Figure 47-4. Endianness Configuration Effect



The size of the payload data to be stored can be selected by configuring SPDIFRX_MR.DATAWIDTH. See the figure below.

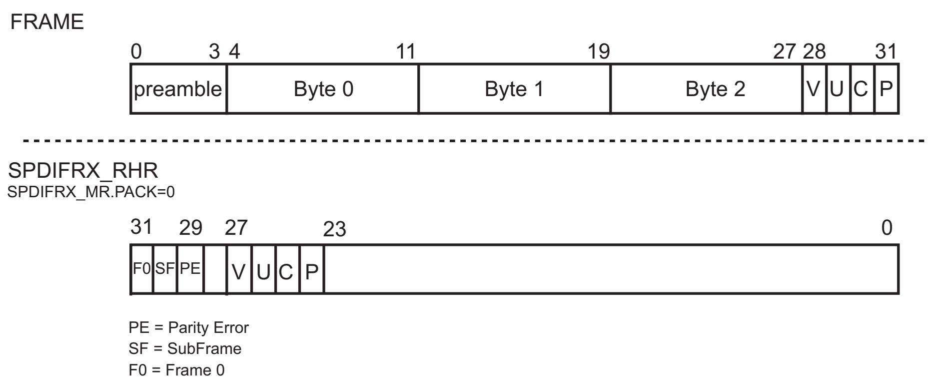
Figure 47-5. Data Size Configuration Effect



When SPDIFRX_MR.PACK is written to '1', only payload data is stored (See FIFO Organization).

When SPDIFRX_MR.PACK is written to '0', the payload data and additional information on the frame are stored in the FIFO. See the figure below.

Figure 47-6. Additional Frame Information



47.6.6 FIFO Organization

The receiver embeds one 32-element FIFO, each element having a 32-bit width. The Receiver Holding register (SPDIFRX_RHR) is the output of the FIFO. Each time a sample is written in the FIFO, the flag SPDIFRX_ISR.RXRDY is set to '1'.

The data organization in the FIFO depends on the configuration of SPDIFRX_MR.PACK and SPDIFRX_MR.DATAWIDTH.

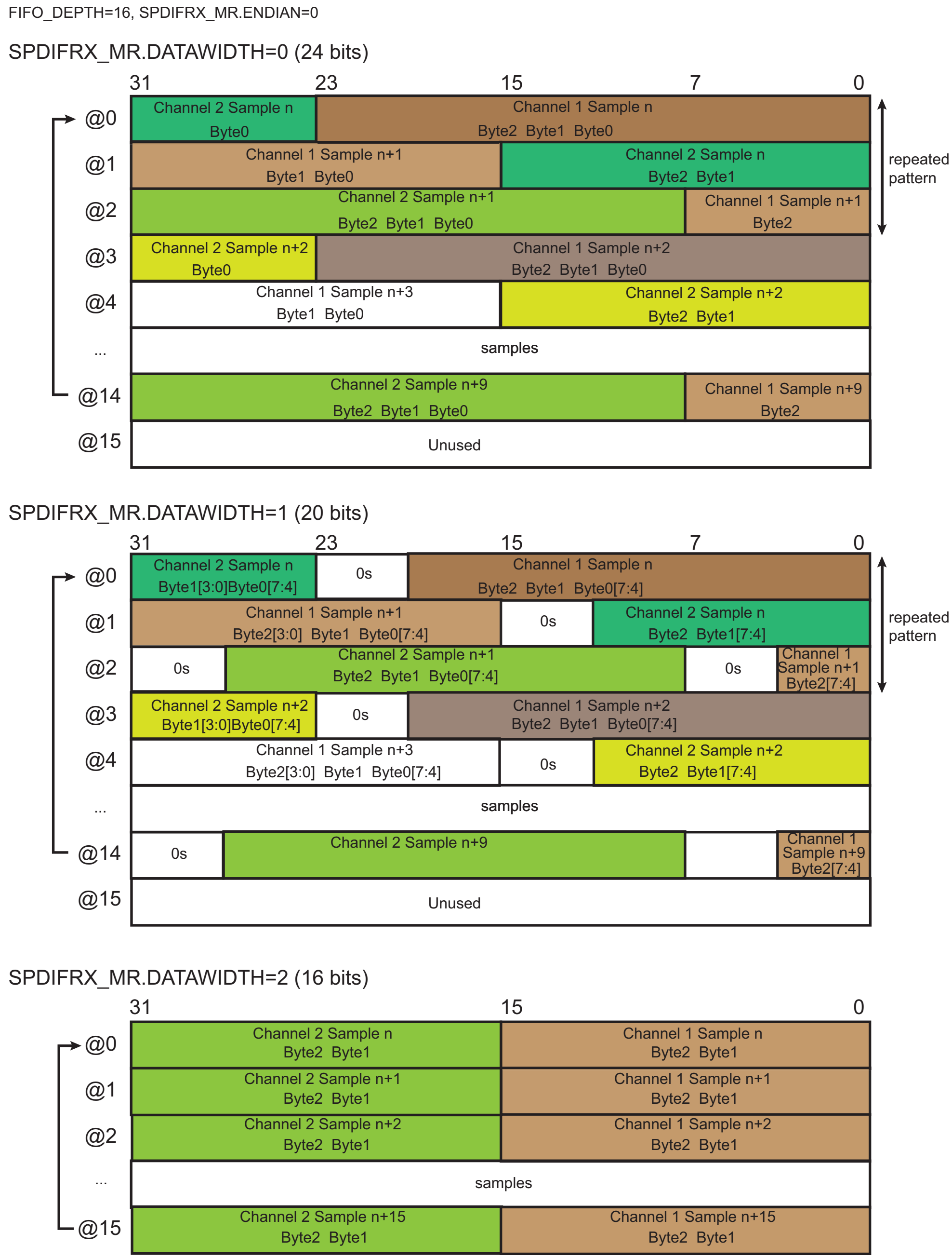
If SPDIFRX_MR.PACK=0, each FIFO element comprises one data sample and additional information (See Data Size Configuration Effect).

If SPDIFRX_MR.PACK=1, there is no additional information associated with each data sample, and the data are packed to optimize system memory amount required for processing. Thus the FIFO organization depends on the value written in the field SPDIFRX_MR.DATAWIDTH. See the figure below.

In 24-bit packed mode, the alignment is always maintained even if an overrun occurs.

When the FIFO is full, SPDIFRX_ISR.RXFULL is set to '1' and an interrupt may be triggered. If an overrun occurs, SPDIFRX_ISR.OVERRUN is set to '1' and an interrupt may also be triggered.

Figure 47-7. FIFO Organization



47.6.7 Channel Status Bit

The channel 1 status bits are stored in the Channel 1 Channel Status register (SPDIFRX_CH1SRx[x=0..5]), a 192-bit register that can be read directly from the user interface. Channel 2 status bits are stored in the Channel 2 Channel Status register (SPDIFRX_CH2SRx[x=0..5]). All 192 bits are valid when the block end status flag rises (SPDIFRX_ISR.BLOCKEND).

SPDIFRX_ISR.C1SC is set to '1' if the channel status bits 0 to 31 differ from two consecutive blocks in channel 1.

SPDIFRX_ISR.C2SC is set to '1' if the channel status bits 0 to 31 differ from two consecutive blocks in channel 2.

47.6.8 User Data Bit

The user data bits of channel 1 are stored in the Channel 1 User Data register (SPDIFRX_CH1UDx[x=0..5]), a 192-bit register that can be read directly from the user interface. The user data bits of channel 2 are stored in the Channel 2 User Data register (SPDIFRX_CH2UDx[x=0..5]). All 192 bits are valid when the block end status flag rises (SPDIFRX_ISR.BLOCKEND).

47.6.9 Write Protection Registers

To prevent any single software error from corrupting SPDIFRX behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SPDIF Receiver Write Protection Mode Register (SPDIFRX_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SPDIF Receiver Write Protection Status Register (SPDIFRX_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SPDIFRX_WPSR.

The following register can be write-protected by setting SPDIFRX_WPMR.WPEN:

- SPDIF Receiver Mode Register

The following registers can be write-protected by setting SPDIFRX_WPMR.WPITEN:

- SPDIF Receiver Interrupt Enable Register
- SPDIF Receiver Interrupt Disable Register

The following register can be write-protected by setting SPDIFRX_WPMR.WPCREN:

- SPDIF Receiver Control Register

47.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SPDIFRX_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	SPDIFRX_MR	31:24								AUTORST
		23:16								
		15:8								SBMODE
		7:0	PACK		DATAWIDTH[1:0]		PBMODE	ENDIAN	VBMODE	RXEN
0x08 ... 0x0F	Reserved									
0x10	SPDIFRX_IER	31:24								
		23:16								
		15:8		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
		7:0	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
0x14	SPDIFRX_IDR	31:24								
		23:16								
		15:8		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
		7:0	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
0x18	SPDIFRX_IMR	31:24								
		23:16								
		15:8		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
		7:0	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
0x1C	SPDIFRX_ISR	31:24								
		23:16								
		15:8		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
		7:0	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
0x20	SPDIFRX_RSR	31:24					IFS[11:8]			
		23:16	IFS[7:0]							
		15:8								
		7:0					NOSIGNAL	LOWF	BADF	ULOCK
0x24	SPDIFRX_RHR	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x28 ... 0x2F	Reserved									
0x30	SPDIFRX_CH1SR0	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x34	SPDIFRX_CH1SR1	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x38	SPDIFRX_CH1SR2	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x3C	SPDIFRX_CH1SR3	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	SPDIFRX_CH1SR4	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x44	SPDIFRX_CH1SR5	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x48	SPDIFRX_CH1UD0	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x4C	SPDIFRX_CH1UD1	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x50	SPDIFRX_CH1UD2	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x54	SPDIFRX_CH1UD3	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x58	SPDIFRX_CH1UD4	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x5C	SPDIFRX_CH1UD5	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x60	SPDIFRX_CH2SR0	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x64	SPDIFRX_CH2SR1	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x68	SPDIFRX_CH2SR2	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x6C	SPDIFRX_CH2SR3	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x70	SPDIFRX_CH2SR4	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							
0x74	SPDIFRX_CH2SR5	31:24	CHSTATUS[31:24]							
		23:16	CHSTATUS[23:16]							
		15:8	CHSTATUS[15:8]							
		7:0	CHSTATUS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	SPDIFRX_CH2UD0	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x7C	SPDIFRX_CH2UD1	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x80	SPDIFRX_CH2UD2	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x84	SPDIFRX_CH2UD3	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x88	SPDIFRX_CH2UD4	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x8C	SPDIFRX_CH2UD5	31:24	CHUSERDATA[31:24]							
		23:16	CHUSERDATA[23:16]							
		15:8	CHUSERDATA[15:8]							
		7:0	CHUSERDATA[7:0]							
0x90 ... 0xE3	Reserved									
0xE4	SPDIFRX_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	SPDIFRX_WPSR	31:24							SWETYP[1:0]	
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0					SWE	SEQE		WPVS

47.7.1 SPDIF Receiver Control Register

Name: SPDIFRX_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the SPDIF Receiver Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the SPDIF receiver.

47.7.2 SPDIF Receiver Mode Register

Name: SPDIFRX_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPDIF Receiver Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								AUTORST
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SBMODE
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	PACK		DATAWIDTH[1:0]		PBMODE	ENDIAN	VBMODE	RXEN
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 24 – AUTORST Consecutive Preamble Error Threshold Automatic Restart

Value	Name	Description
0	NOACTION	No action whatever the number of consecutive preamble errors found during the period where SPDIFRX_RSR.ULOCK=0.
1	UNLOCK_ON_PRE_ERR	If 16 consecutive preamble errors are detected, the clock recovery circuitry is restarted.

Bit 8 – SBMODE Start of Block Bit Mode

Value	Name	Description
0	ALWAYS_LOAD	Whatever the preamble code, the sample is loaded in FIFO.
1	DISCARD	The sample is loaded in FIFO only if a Start of Block is detected.

Bit 7 – PACK Packed Data Mode in Receive Holding Register

Value	Name	Description
0	DISABLED	Each read of SPDIFRX_RHR contains 1 sample and additional information (validity bit, parity bit, user data bit, channel status bit, byte 0 bit 1 of channel status and type of frame carrying the sample).
1	ENABLED	The 32-bit SPDIFRX_RHR contains only payload data. Depending on the value of SPDIFRX_MR.DATAWIDTH, the alignment of data differs. This mode optimizes the amount of system memory required to manage the samples.

Bits 5:4 – DATAWIDTH[1:0] Sample Data Width

Value	Name	Description
0	24BIT	The complete data field is stored in FIFO.
1	20BIT	Only the 20 MSB are stored in the FIFO.
2	16BIT	Only the 16 MSB are stored in the FIFO.
3	Reserved	Reserved

Bit 3 – PBMODE Parity Bit Mode

Value	Name	Description
0	PARCHECK	Parity check enabled on data payload.
1	NOPARCHECK	There is no parity check on data payload.

Bit 2 – ENDIAN Data Word Endian Mode

Value	Name	Description
0	LITTLE	Little-endian mode for 24-bit samples.
1	BIG	Big-endian mode for 24-bit samples.

Bit 1 – VBMODE Validity Bit Mode

Value	Name	Description
0	ALWAYS_LOAD	Whatever the validity bit value is, the sample is loaded in FIFO.
1	DISCARD_IF_VB1	The sample is loaded in FIFO only if the validity bit equals 0.

Bit 0 – RXEN SPDIF Receive Enable

Value	Name	Description
0	DISABLE	SPDIF receiver is disabled.
1	ENABLE	SPDIF receiver is enabled.

47.7.3 SPDIF Receiver Interrupt Enable Register

Name: SPDIFRX_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPDIF Receiver Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 14 – CP_ERR 16 Consecutive Preamble Errors Interrupt Enable

Bit 13 – PRE_ERR Preamble Error (Code Violation) Enable

Bit 12 – NRZ_ERR NRZ Biphase Mark Error in Payload Data (Code Violation) Enable

Bit 11 – BLOCKST Start of Block Interrupt Enable

Bit 10 – SECE Security Report Interrupt Enable

Bit 9 – C2SC Bit 0 to 31 Channel 2 Status Change Interrupt Enable

Bit 8 – C1SC Bit 0 to 31 Channel 1 Status Change Interrupt Enable

Bit 7 – RXFULL Receiver FIFO Full Interrupt Enable

Bit 6 – OVERRUN FIFO Overrun, Interrupt Enable

Bit 5 – PAR_ERR Parity Bit Error Interrupt Enable

Bit 4 – SFE Sampling Frequency Change Event Interrupt Enable

Bit 3 – BLOCKEND End of Block Interrupt Enable

Bit 2 – LOSS Loss of Signal Activity While Locked Interrupt Enable

Bit 1 – LOCKED Receiver Synchronized Interrupt Enable

Bit 0 – RXRDY Receive Data Ready Interrupt Enable

47.7.4 SPDIF Receiver Interrupt Disable Register

Name: SPDIFRX_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPDIF Receiver Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 14 – CP_ERR 16 Consecutive Preamble Errors Interrupt Enable

Bit 13 – PRE_ERR Preamble Error (Code Violation) Disable

Bit 12 – NRZ_ERR NRZ Biphase Mark Error in Payload Data (Code Violation) Disable

Bit 11 – BLOCKST Start of Block Interrupt Disable

Bit 10 – SECE Security Report Interrupt Disable

Bit 9 – C2SC Bit 0 to 31 Channel 2 Status Change Interrupt Disable

Bit 8 – C1SC Bit 0 to 31 Channel 1 Status Change Interrupt Disable

Bit 7 – RXFULL Receiver FIFO Full Interrupt Disable

Bit 6 – OVERRUN FIFO Overrun, Interrupt Disable

Bit 5 – PAR_ERR Parity Bit Error Interrupt Disable

Bit 4 – SFE Sampling Frequency Change Event Interrupt Disable

Bit 3 – BLOCKEND End of Block Interrupt Disable

Bit 2 – LOSS Loss of Signal Activity While Locked Interrupt Disable

Bit 1 – LOCKED Receiver Synchronized Interrupt Disable

Bit 0 – RXRDY Receive Data Ready Interrupt Disable

47.7.5 SPDIF Receiver Interrupt Mask Register

Name: SPDIFRX_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
Access		W	W	W	W	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
Access	R	R	R	W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 14 – CP_ERR 16 Consecutive Preamble Errors Interrupt Enable

Bit 13 – PRE_ERR Preamble Error (Code Violation) Interrupt Mask

Bit 12 – NRZ_ERR NRZ Biphase Mark Error in Payload Data (Code Violation) Interrupt Mask

Bit 11 – BLOCKST Start of Block Interrupt Mask

Bit 10 – SECE Security Report Interrupt Mask

Bit 9 – C2SC Bit 0 to 31 Channel 2 Status Change Interrupt Mask

Bit 8 – C1SC Bit 0 to 31 Channel 1 Status Change Interrupt Mask

Bit 7 – RXFULL Receiver FIFO Full Interrupt Mask

Bit 6 – OVERRUN FIFO Overrun, Interrupt Mask

Bit 5 – PAR_ERR Parity Bit Error Interrupt Mask

Bit 4 – SFE Sampling Frequency Change Event Interrupt Mask

Bit 3 – BLOCKEND End of Block Interrupt Mask

Bit 2 – LOSS Loss of Signal Activity While Locked Interrupt Mask

Bit 1 – LOCKED Receiver Synchronized Interrupt Mask

Bit 0 – RXRDY Receive Data Ready Interrupt Mask

47.7.6 SPDIF Receiver Interrupt Status Register

Name: SPDIFRX_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		CP_ERR	PRE_ERR	NRZ_ERR	BLOCKST	SECE	C2SC	C1SC
Access		W	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXFULL	OVERRUN	PAR_ERR	SFE	BLOCKEND	LOSS	LOCKED	RXRDY
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 14 – CP_ERR 16 Consecutive Preamble Errors Interrupt Enable

Bit 13 – PRE_ERR Preamble Error (code violation) Status (cleared on read)

Bit 12 – NRZ_ERR NRZ Biphase Mark Error in payload data (code violation) Status (cleared on read)

Bit 11 – BLOCKST Start of Block Interrupt Status (cleared on read)

Bit 10 – SECE Security Report Interrupt Status (cleared on read)
This flag is set when either WPVS, SEQE or SWE is set in SPDIFRX_WPSR.

Bit 9 – C2SC Bit 0 to 31 Channel 2 Status Change (cleared on read)

Bit 8 – C1SC Bit 0 to 31 Channel 1 Status Change (cleared on read)

Bit 7 – RXFULL Receiver FIFO Full Interrupt Status (cleared on read)

Bit 6 – OVERRUN FIFO Overrun, Interrupt Status (cleared on read)

Bit 5 – PAR_ERR Parity Bit Error Interrupt Status (cleared on read)

Bit 4 – SFE Sampling Frequency Change Event Interrupt Status (cleared on read)

Bit 3 – BLOCKEND End of Block Interrupt Status (cleared on read)

Bit 2 – LOSS Loss of Signal Activity While Locked Interrupt Status (cleared on read)

Bit 1 – LOCKED Receiver Synchronized Interrupt Status (cleared on read)

Bit 0 – RXRDY Receive Data Ready Interrupt Status (cleared when reading SPDIFRX_RHR)

47.7.7 SPDIF Receiver Status Register

Name: SPDIFRX_RSR
Offset: 0x20
Reset: 0x00000001
Property: Read-only

Note: This status register is not cleared on read; it reports the current state of the receiver.

Bit	31	30	29	28	27	26	25	24
					IFS[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IFS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					NOSIGNAL	LOWF	BADF	ULOCK
Access					R	R	R	R
Reset					0	0	0	1

Bits 27:16 – IFS[11:0] Image of Sampling Frequency
When ULOCK=0, IFS returns the number of GCLK periods for two symbols. This field can be used to approximate the sample frequency (f_S).

$$f_S\ (\mathrm{kHz})\ \approx \frac{f_{\mathrm{GCLK}}\ (\mathrm{kHz})}{32 \times \mathrm{IFS}}$$

with IFS in decimal.

Bit 3 – NOSIGNAL No Signal on Receive Line

Value	Description
0	The receiver is synchronized or searching for receive line frequency or preambles.
1	The receiver is not able to find any activity (no edge) on RX line.

Bit 2 – LOWF Low Clock Frequency Provided on Peripheral GCLK Clock

Value	Description
0	The receiver is synchronized or the receiver is determining the sample frequency of the receive line or searching preambles.
1	The receiver is not able to recover the protocol because the GCLK clock frequency is lower than the minimum required.

Bit 1 – BADF Bad Format Detected on SPDIF RX Line

Value	Description
0	The receiver is synchronized or the receiver is searching for SPDIF receive line frequency or preambles.
1	The receiver is not able to detect a SPDIF format on the receive line.

Bit 0 – ULOCK Unlocked Receiver

Value	Description
0	The receiver is synchronized and loads data.

Value	Description
1	The receiver is not locked because it is searching for SPDIF receive line frequency or preambles. See BADF, LOWF, and NOSIGNAL for reasons of unlocked state.

47.7.8 SPDIF Receiver Holding Register

Name: SPDIFRX_RHR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Channel 1 and 2 Data
The data format depends on the configuration of SPDIFRX_MR.DATAWIDTH and SPDIFRX_MR.PACK. See Clock and Data Recovery.

47.7.9 SPDIF Receiver Channel 1 Channel Status Register

Name: SPDIFRX_CH1SRx
Offset: 0x30 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CHSTATUS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHSTATUS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHSTATUS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHSTATUS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHSTATUS[31:0] Channel 1 Status Data
Channel 1 status bits.
The complete status channel is 192 bits long.
Channel 1 status bits 31 down to 0 are loaded into SPDIFRX_CH1SR0[31:0], channel 1 status bits 63 down to 32 are loaded into SPDIFRX_CH1SR1[31:0], etc.

47.7.10 SPDIF Receiver Channel 1 User Data Register

Name: SPDIFRX_CH1UDx
Offset: 0x48 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CHUSERDATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHUSERDATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHUSERDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHUSERDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHUSERDATA[31:0] Channel 1 User Data
Channel 1 user data bits.
The complete user data is 192 bits long.
Channel 1 user data bits 31 down to 0 are loaded into SPDIFRX_CH1UD0[31:0], channel 1 user data bits 63 down to 32 are loaded into SPDIFRX_CH1UD1[31:0], etc.

47.7.11 SPDIF Receiver Channel 2 Channel Status Register

Name: SPDIFRX_CH2SRx
Offset: 0x60 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CHSTATUS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHSTATUS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHSTATUS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHSTATUS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHSTATUS[31:0] Channel 2 Status Data
Channel 2 status bits.
The complete status channel is 192 bits long.
Channel 2 status bits 31 down to 0 are loaded into SPDIFRX_CH2SR0[31:0], channel 2 status bits 63 down to 32 are loaded into SPDIFRX_CH2SR1[31:0], etc.

47.7.12 SPDIF Receiver Channel 2 User Data Register

Name: SPDIFRX_CH2UDx
Offset: 0x78 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CHUSERDATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHUSERDATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHUSERDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHUSERDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHUSERDATA[31:0] Channel 2 User Data
Channel 2 User Data bits.
The complete user data is 192 bits long.
Channel 2 user data bits 31 down to 0 are loaded into SPDIFRX_CH2UD0[31:0], channel 2 user data bits 63 down to 32 are loaded into SPDIFRX_CH2UD1[31:0], etc.

47.7.13 SPDIF Receiver Write Protection Mode Register

Name: SPDIFRX_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535044	PASSWD	Writing any other value in this field aborts the write operation of the WPEN, WPITEN and WPCREN bits. Always reads at 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in SPDIFRX_WPSR.WPSRC and the last software control error type is reported in SPDIFRX_WPSR.SWETYP. The SPDIFRX_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in SPDIFRX_WPSR.WPSRC and only the first software control error type is reported in SPDIFRX_WPSR.SWETYP. The SPDIFRX_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register (SPDIFRX_CR) if WPKEY corresponds to 0x535044.
1	Enables the write protection on the Control register (SPDIFRX_CR) if WPKEY corresponds to 0x535044.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to 0x535044.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x535044.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535044 (“SPD” in ASCII).

Value	Description
1	Enables the write protection is enabled. All accesses to configuration registers are canceled and generate an error in the SPDIFRX_WPSR register.

47.7.14 SPDIF Receiver Write Protection Status Register

Name: SPDIFRX_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							SWETYP[1:0]	
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE		WPVS
Access					R	R		R
Reset					0	0		0

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	UNDEF_RW	Access to an undefined address (warning)

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SPDIFRX_WPSR.
1	A software error has occurred since the last read of SPDIFRX_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of SPDIFRX_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of SPDIFRX_WPSR. This flag can only be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of SPDIFRX_WPSR.
1	A write protection violation has occurred since the last read of SPDIFRX_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

48. Sony/Philips Digital Interface Transmitter (SPDIFTX)

48.1 Description

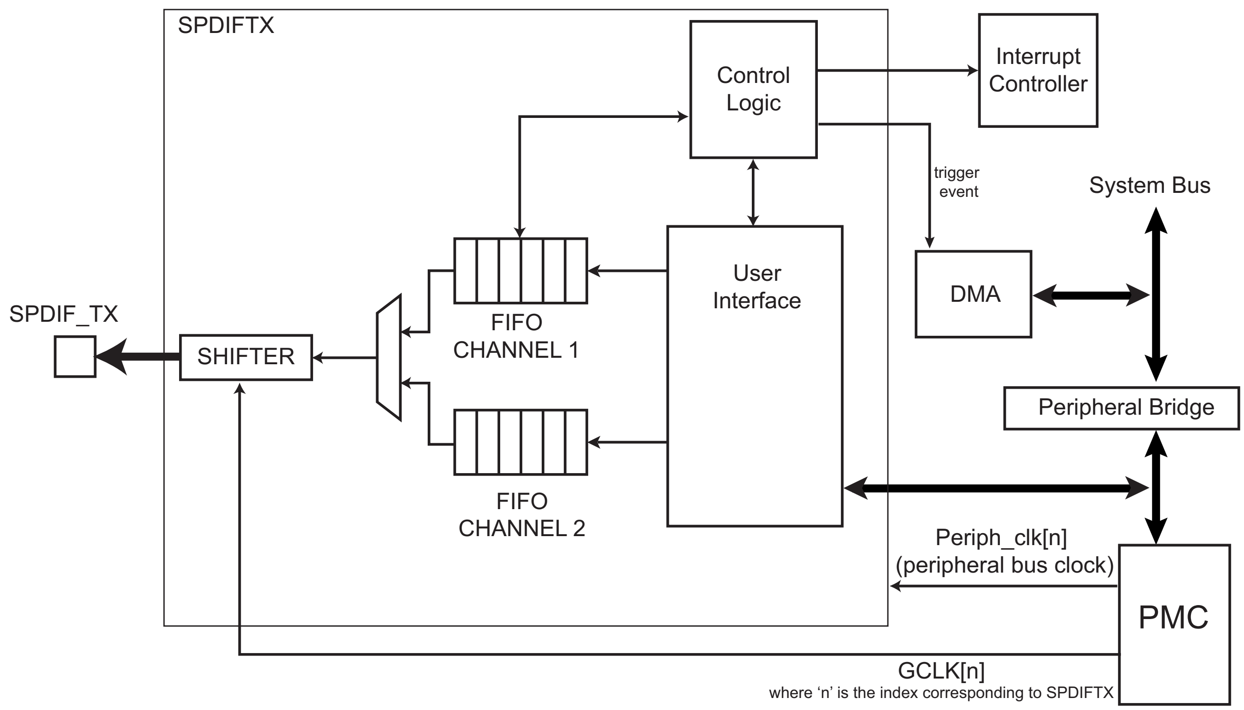
The Sony/Philips Digital Interface Transmitter (SPDIFTX) is a serial port compliant with the IEC-60958 standard.

48.2 Embedded Characteristics

- SPDIF Compatible Serial Port
- Programmable User Data Field
- Programmable Channel Status Field

48.3 Block Diagram

Figure 48-1. SPDIFTX Block Diagram



48.4 Signal Description

Table 48-1. Pin Description

Pin Name	Description	Direction
SPDIF_TX	SPDIF Output Port	Output

48.5 Product Dependencies

48.5.1 I/O Lines

The pins used for interfacing the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the SPDIFTX pins to their peripheral functions.

48.5.2 Power Management

The SPDIFTX is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC before using the SPDIFTX.

48.5.3 Interrupt Sources

The SPDIFTX interface has an interrupt line connected to the interrupt controller. Handling the SPDIFTX interrupt requires programming the interrupt controller before configuring the SPDIFTX.

48.6 Functional Description

48.6.1 SPDIF Transmitter Transmission

48.6.1.1 SPDIF Protocol

Data transmission complies with the IEC-60958 standard.

Data transmission is performed by the transfer of blocks of 192 frames.

Each frame is divided into two subframes, one for each channel.

Each subframe is divided into 32 time slots carrying the following fields:

- Preamble (4 bits)–There are three possible preambles named B, M and W. Preambles are specific patterns providing synchronization and identification of the subframes and blocks.
- Data (24 bits)–This field can carry a PCM audio data or any other information.
- Validity bit–This bit can be written to ‘0’ or to ‘1’ by the user with the VALIDx bit in the Mode register (SPDIFTX_MR).
- User data bit–User data may be used in any way required by the user. It is organized in a 192-bit block.
- Channel status bit–For every subframe, the channel status provides information related to the data carried in the main data field of that same subframe. It is organized in a 192-bit block.
- Parity bit–This bit ensures that time slots 4 to 31 carry an even number of ones and an even number of zeros.

Figure 48-2. SPDIF Transmitter Frame and Block Format

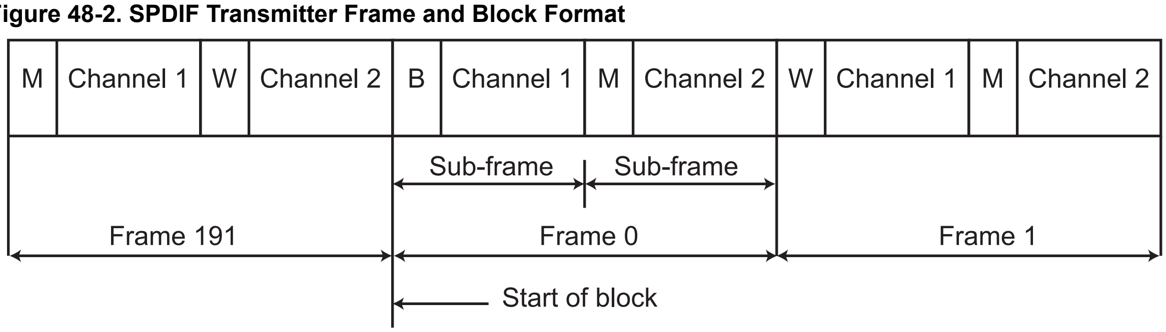
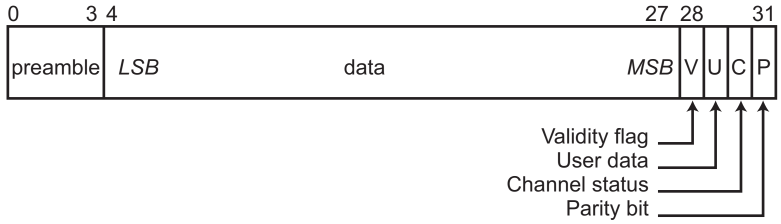


Figure 48-3. SPDIF Transmitter Subframe Format



48.6.1.2 SPDIF Transmitter Data Rate

The SPDIFTX sends symbols at the rate of GCLK. Thus the data rate is managed by the GCLK clock generated by the PMC.

Each bit to send is converted into 2 symbols. Thus a frame has a length of 128 symbols. In order to respect a sample rate of F_S, the frequency of GCLK must be F_{GCLK}=128 × F_S. GCLK must be derived from an audio clock.

48.6.2 SPDIF Transmitter FIFO

The controller embeds one 32-element FIFO per channel. The FIFO is organized in bytes so that the selected data format is optimized.

The status of the FIFOs is reported in the Interrupt Status register (SPDIFTX_ISR).

The FIFO can be filled while the SPDIFTX is disenabled.

The status flags of the two FIFOs are merged into one status register. Thus, a ready flag is raised only if the two FIFOs are ready. An error flag is raised if at least one of the two FIFOs has an error.

48.6.3 Data Organization

The data field defined by the IEC-60958 standard is 3 bytes long (24 bits). All data to be transmitted on the SPDIF line is sent through the Common Data register (SPDIFTX_CDR).

Data is organized using the following fields in SPDIFTX_MR:

- MULTICH–defines the number of channels that are sent to the interface. When MULTICH is configured in mono channel, the data written in SPDIFTX_CDR is duplicated on the two SPDIF channels. When MULTICH is configured in Dual Channel mode, each SPDIF channel has its own data stream.
- ENDIAN–defines the byte organization in SPDIFTX_CDR (big-endian or little-endian).
- JUSTIFY–defines the location of the valid bits in the CDR/memory container (MSB or LSB). If the data is LSB-justified, the controller automatically shifts the data to align on SPDIF.
- BPS–defines a container of data for one data (it defines the number of valid bytes in SPDIFTX_CDR) and ranges from 1 to 4 bytes.
- VBPS–defines the number of valid bits in the container and ranges from 1 to 32.
- CMODE–defines the transfer mode when BPS is defined on 3 bytes.

48.6.3.1 Automatic Data Alignment

When the number of valid bits in the data to be transmitted is smaller than the size of the container (VPBS < 8*(BPS+1)), the SPDIFTX must be instructed with the justification of the received data (data written in SPDIFTX_CDR).

SPDIFTX_MR.JUSTIFY defines the alignment of the valid bits per sample (defined by VPBS) of the data written in SPDIFTX_CDR.CDR.

If the valid bits are MSB-aligned in the container, SPDIFTX_MR.JUSTIFY=1.

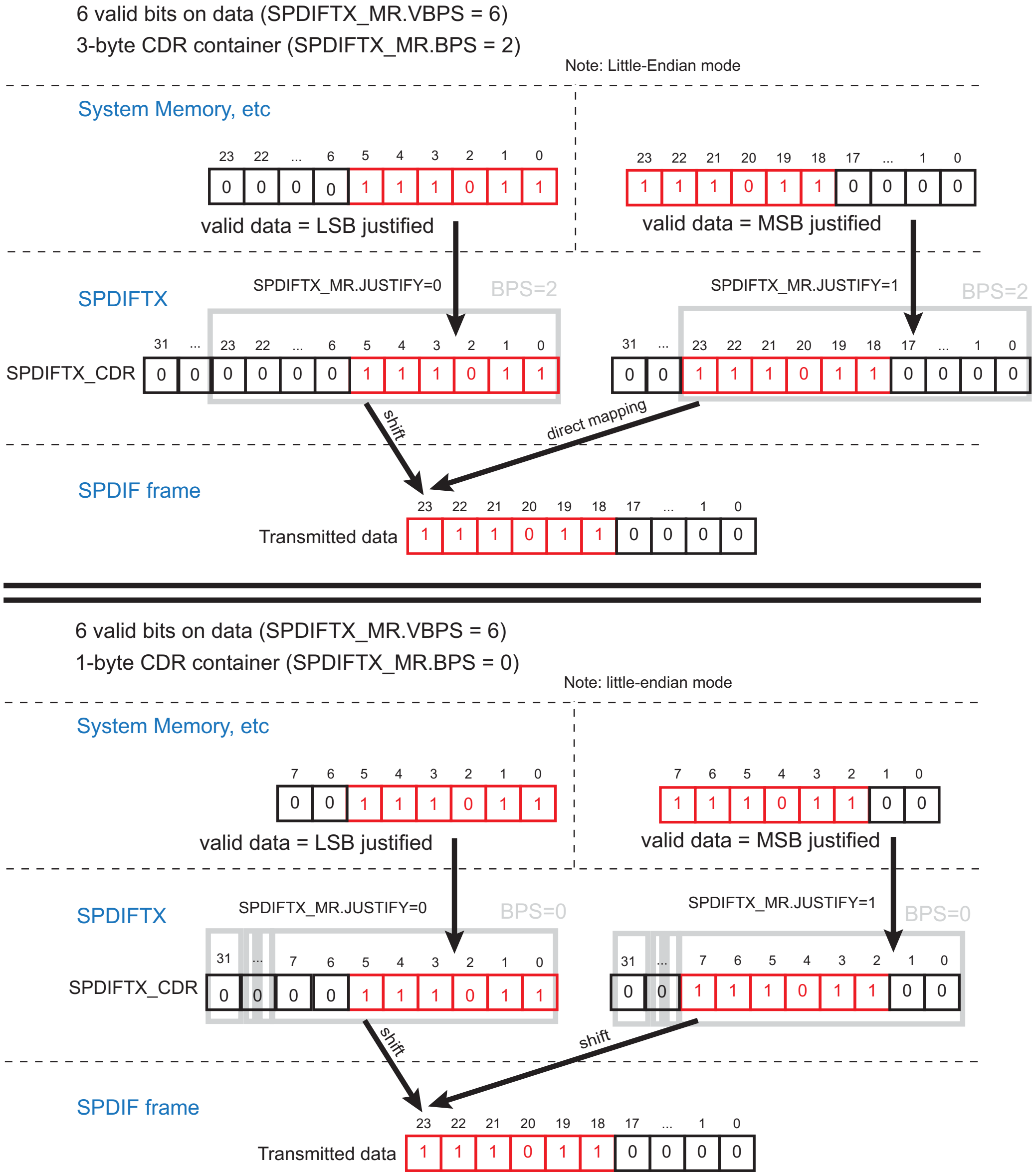
If the valid bits are LSB-aligned in the container, SPDIFTX_MR.JUSTIFY=0.

When necessary, the SPDIFTX automatically realigns CDR data. The automatic realignment prevents software intervention or manipulation on the data buffer prior to writing the SPDIFTX_CDR.

There is no alignment when the number of valid bits per sample configured in VPBS is equal to the size of the container configured in BPS (VPBS = 8*(BPS+1)).

VBPS must be less than or equal to 8 times the number of bytes defined in BPS (VBPS ≤ 8*(BPS+1)).

Figure 48-4. Data Justification Example



48.6.3.2 Up to 8-bit Data

When SPDIFTX_MR.BPS=0, each write access to SPDIFTX_CDR provides up to four data samples.

When the size of the data to transmit is lower than or equal to 8 bits, this operating mode reduces the system bus bandwidth required to transfer the buffer of data.

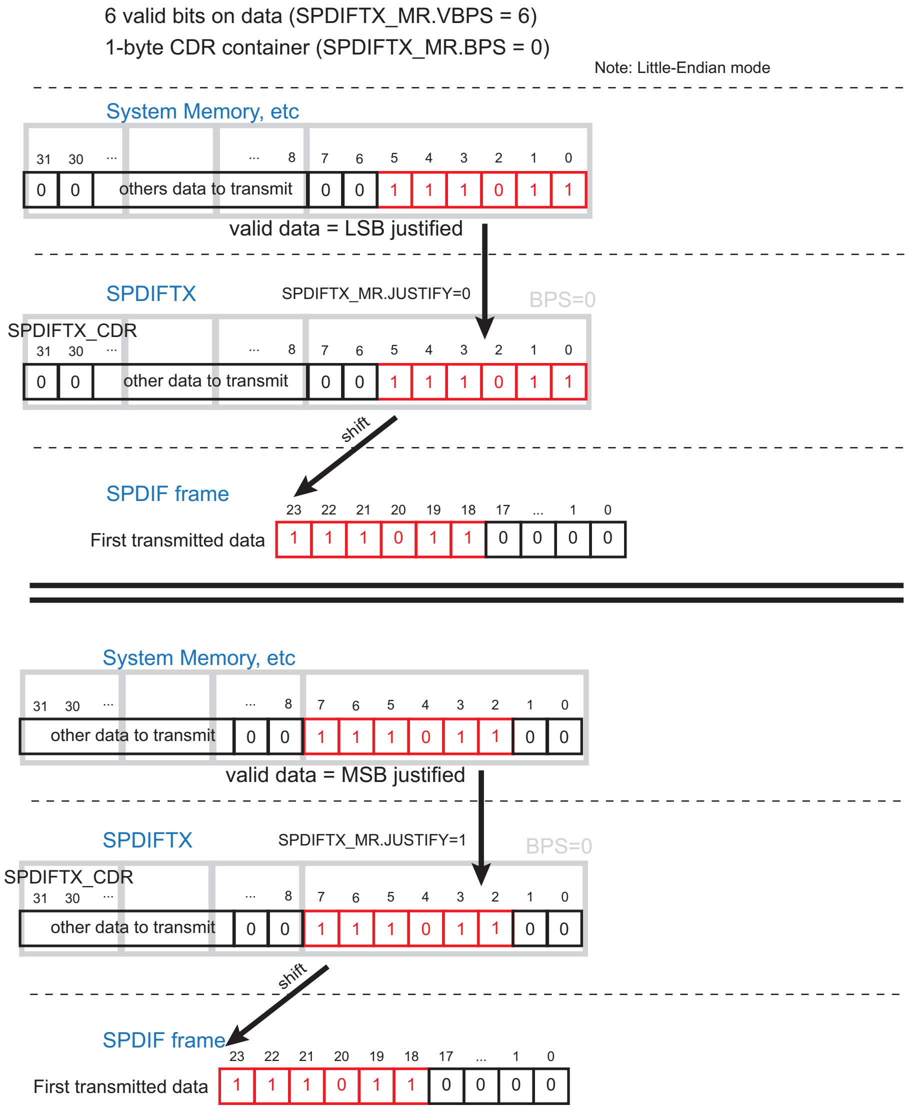
When SPDIFTX_MR.MULTICH=1 (stereo channels, etc.), two data (e.g. 2 stereo audio samples) are written per access to SPDIFTX_CDR. When SPDIFTX_MR.MULTICH=0 (MONO), four data (e.g. 4 mono audio samples) are written per SPDIFTX_CDR access.

When SPDIFTX_MR.BPS=0, the SPDIFTX_MR.ENDIAN bit has no effect.

The first byte sent is located on the LSB part of the SPDIFTX_CDR register.

When necessary, the SPDIFTX_CDR byte to send is MSB-aligned on the 24-bit data word transmitted on the SPDIF line.

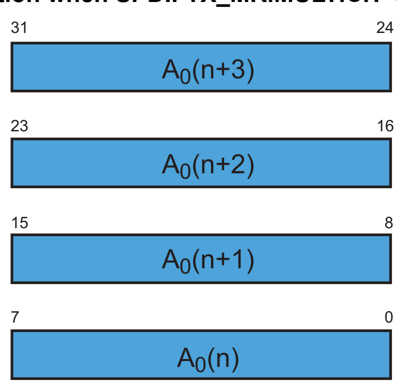
Figure 48-5. Transmitting Byte-Oriented Data Carrying 6 Valid Bits



In the following figures, A_x represents the x^{th} byte of channel 1 and A_0 is the least significant byte of A. B_x represents the x^{th} byte of channel 2 and B_0 is the least significant byte of B.

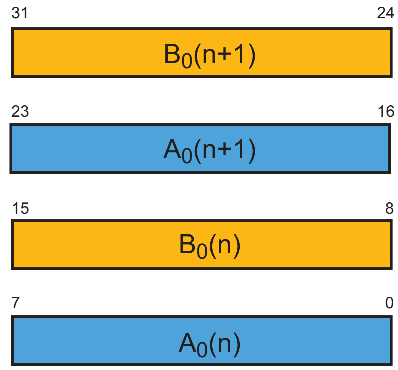
When SPDIFTX_MR.MULTICH=0, each byte of channel 1 is duplicated on channel 2

Figure 48-6. SPDIFTX_CDR Organization when SPDIFTX_MR.MULTICH=0



In SPDIFTX_MR.MULTICH=1, bytes are sent alternately to channel 1 and channel 2.

Figure 48-7. SPDIFTX_CDR Organization when SPDIFTX_MR.MULTICH=1



48.6.3.3 9-bit to 16-bit Data

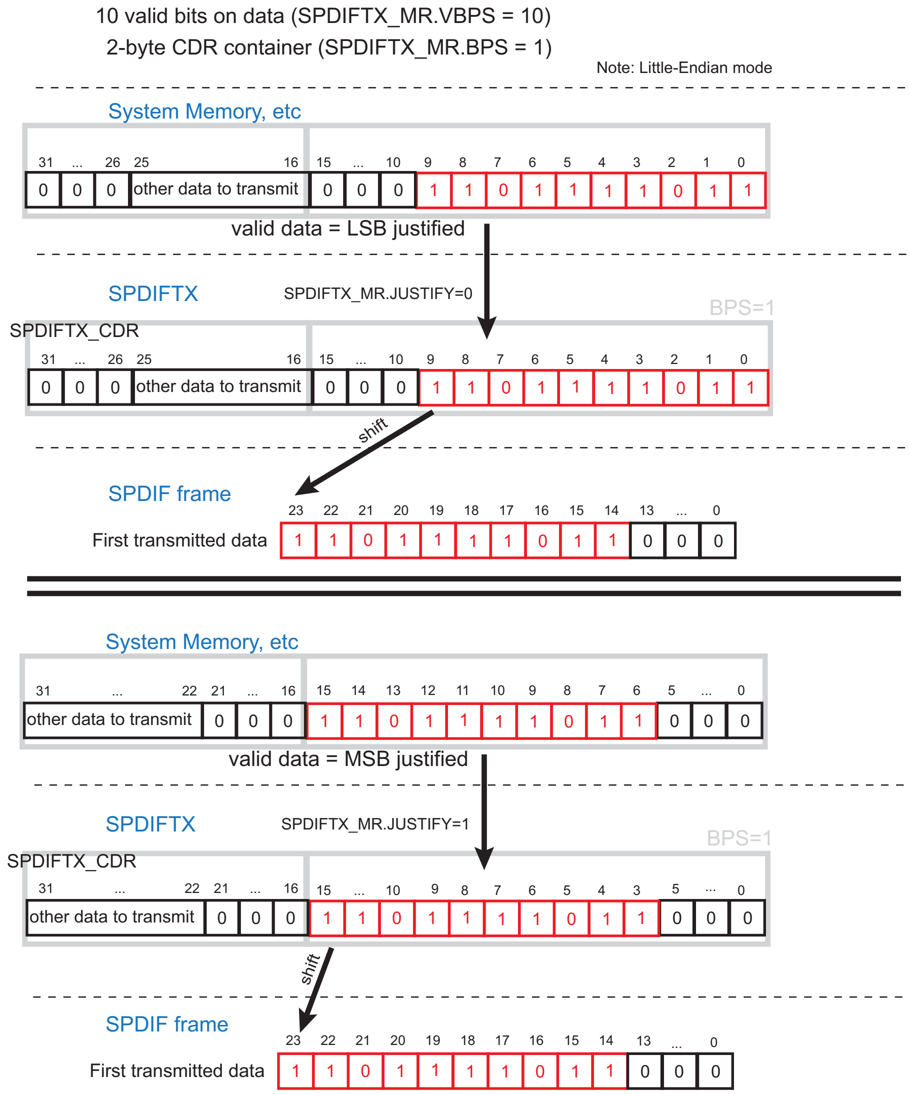
When SPDIFTX_MR.BPS=1, each write access to SPDIFTX_CDR provides two data samples.

When the size of the data to transmit is lower than or equal to 16 bits, this operating mode reduces the system bus bandwidth required to transfer the buffer of data.

The data of channel 1 is located in SPDIFTX_CDR.CDR[15:0]. SPDIFTX_CDR.CDR[31:0] are loaded by either the next data to send (mono channel) or the data of channel 2 (stereo channels).

When necessary, the SPDIFTX_CDR half-word to send is MSB-aligned on the 24-bit data word transmitted on the SPDIF line.

Figure 48-8. Transmitting Halfword-Oriented Data Carrying 10 Valid Bits



In the following figures, A_x represents the x^{th} byte of channel 1 and A_0 is the least significant byte of A. B_x represents the x^{th} byte of channel 2 and B_0 is the least significant byte of B.

The byte organization of data depends on the configuration of SPDIFTX_MR.ENDIAN.

Figure 48-9. SPDIFTX_CDR Organization for BYTE2, Mono Mode

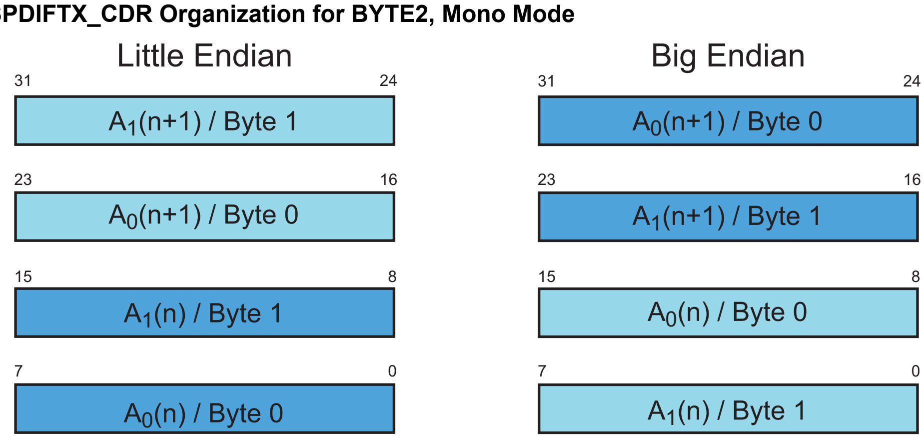
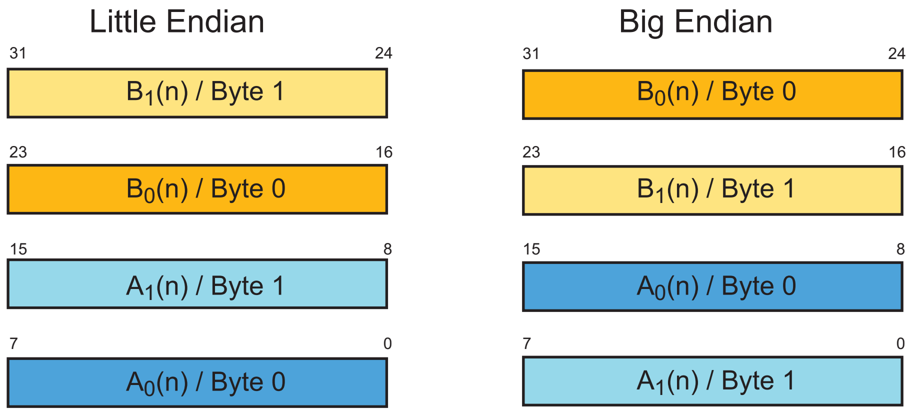


Figure 48-10. SPDIFTX_CDR Organization for BYTE2, Dual Mode



48.6.3.4 17-bit to 24-bit Data

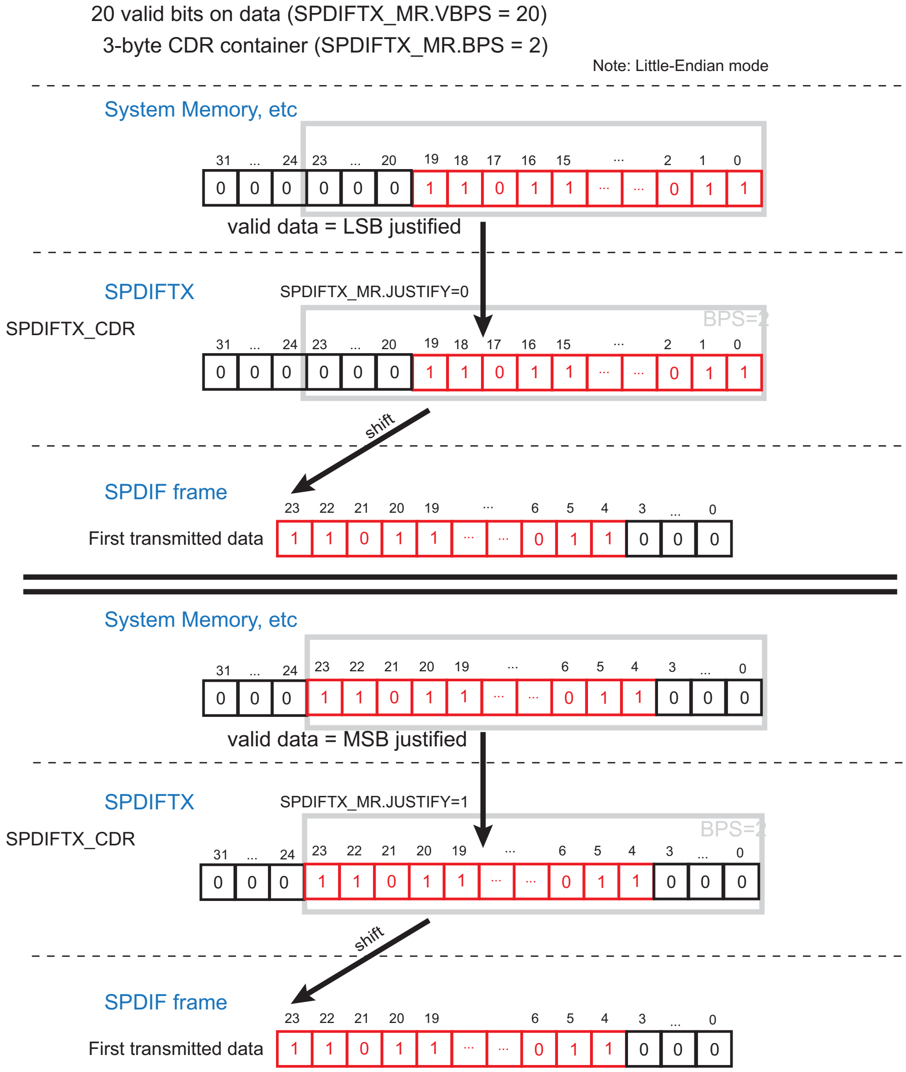
When SPDIFTX_MR.BPS=2, the data organization depends on the value of SPDIFTX_MR.CMODE.

When the size of the data to transmit is lower than or equal to 24 bits, this operating mode reduces the system bus bandwidth required to transfer the buffer of data.

When CMODE=0, the destination of the data in SPDIFTX_CDR depends on the value of the MSB [31:24] of SPDIFTX_CDR.CDR. If CDR[31:24]=0, the data is sent to channel 1. If CDR[31:24]=1, the data is sent to channel 2.

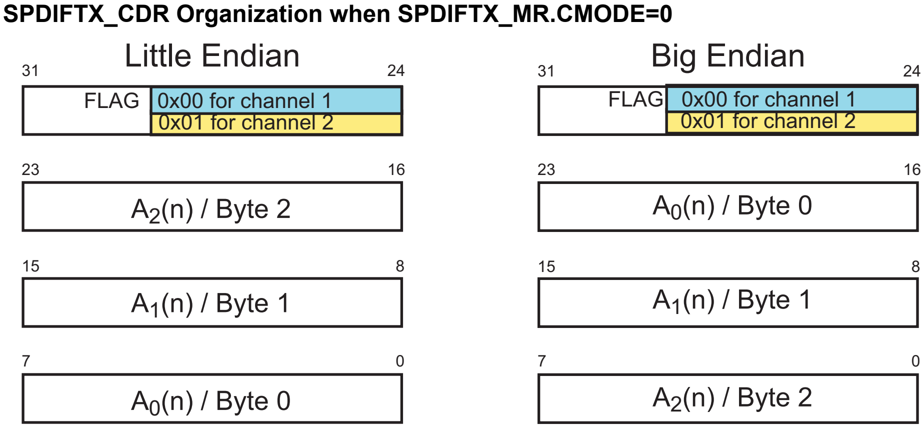
The 3-byte justified data written in SPDIFTX_CDR.CDR[23:0] is transmitted on the SPDIF line.

Figure 48-11. Transmitting 24-bit Oriented Data Carrying 20 Valid Bits



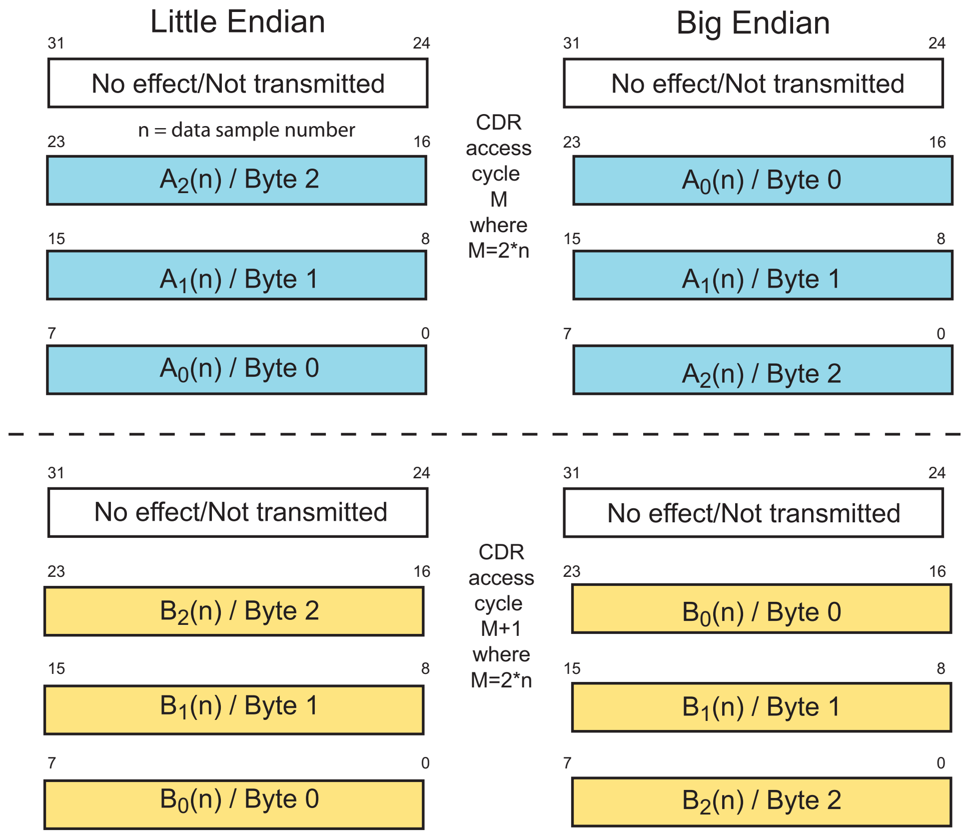
In the following figures, A_x represents the x^{th} byte of channel 1 and A_0 is the least significant byte of A. B_x represents the x^{th} byte of channel 2 and B_0 is the least significant byte of B.

Figure 48-12. SPDIFTX_CDR Organization when SPDIFTX_MR.CMODE=0



When SPDIFTX_MR.CMODE=1, data are sent alternately on channel 1 and channel 2. The first sent data is sent to channel 1 (even cycles), the following data is sent to channel 2 (odd cycles).

Figure 48-13. SPDIFTX_CDR Register Organization when SPDIFTX_MR.CMODE=1

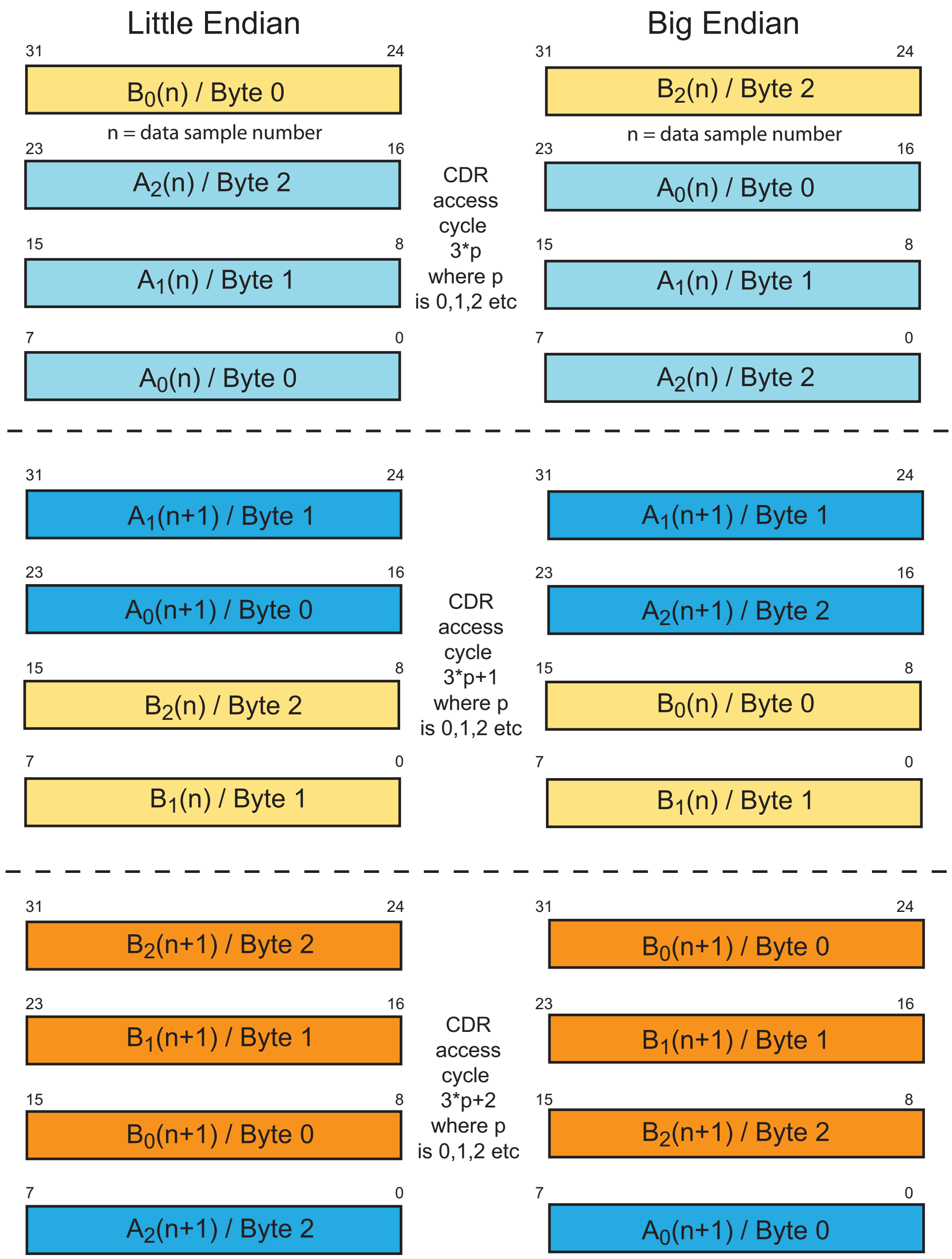


When SPDIFTX_MR.CMODE=2, the write access to SPDIFTX_CDR is optimized for 24-bit data samples being compacted on 32 bits.

- The first access to SPDIFTX_CDR contains the data for channel 1 on CDR[23:0]. CDR[31:24] contain part of the data for channel 2.
- The second access contains the remaining data of channel 2 (CDR[15:0]). CDR[31:16] contain part of the data to send to channel 1.
- The third access contains CDR[7:0] remaining data for channel 1. CDR[31:8] contain the data to be sent to channel 2.

The total size of the data to send must be an integer multiple of 3 so that there is no interference between channels.

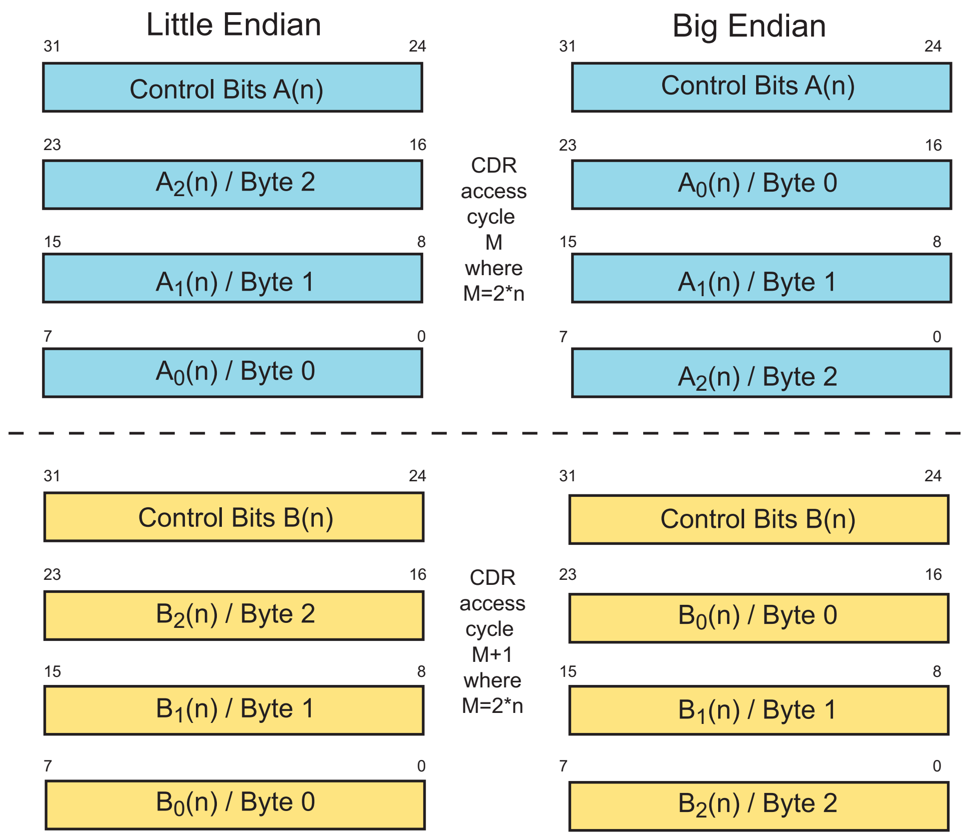
Figure 48-14. SPDIFTX_CDR Organization when SPDIFTX_MR.CMODE=2



When SPDIFTX_MR.CMODE=3, SPDIFTX_CDR.CDR[23:0] contains only one 24-bit data and CDR[31:24] contains SPDIF frame control bits.

Depending on the configuration of SPDIFTX_EMR, the control bits supplied in the MSB byte are used or not. When SPDIFTX_MR.CMODE=3, the User Data, Channel Status, validity bit and parity error are managed on a per data basis. If SPDIFTX_EMR.PCM is set, the preamble is no longer set automatically by the SPDIFTX; the preamble type is selected using SPDIFTX_CDR.PC instead.

Figure 48-15. SPDIFTX_CDR Organization when SPDIFTX_MR.CMODE=3 Mode

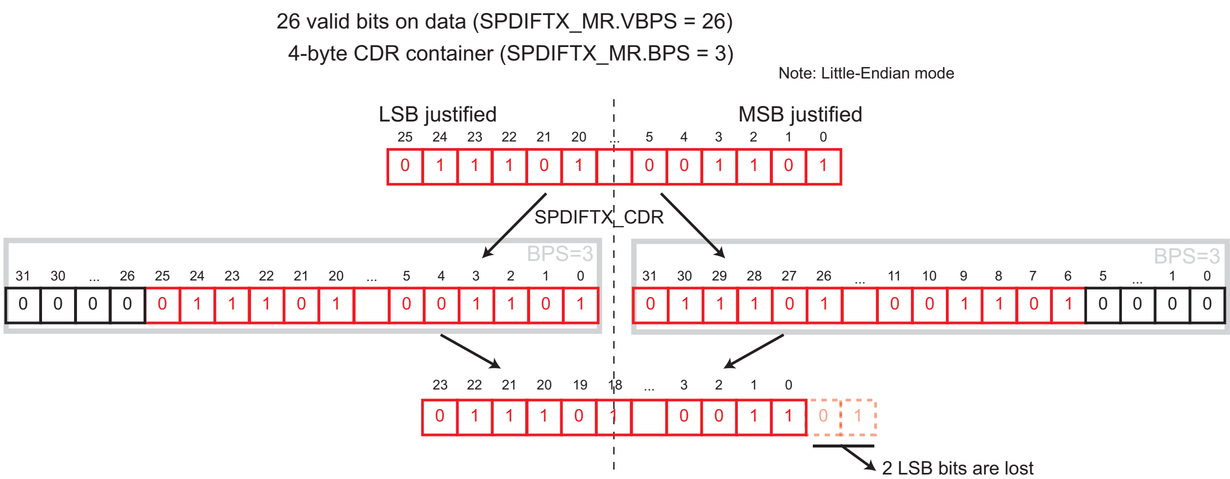


48.6.3.5 25-bit to 32-bit Data

When SPDIFTX_MR.BPS=3, SPDIFTX_MR.CMODE must be configured to 3.

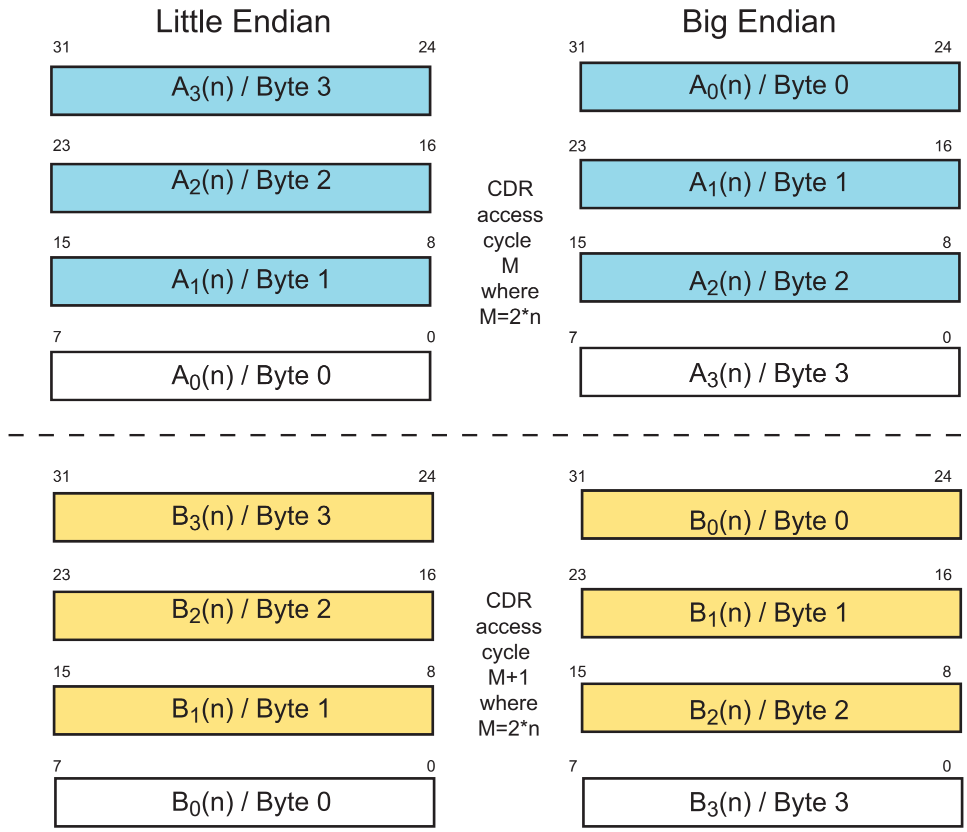
Due to the 24-bit data definition of the IEC standard, the data written in SPDIFTX_CDR is first shifted by 8 bits before being sent. In this mode, VPBS must be at least equal to 24.

Figure 48-16. Sending a 4-Byte Data Defined on 26 Bits



In the following figures, A_x represents the x^{th} byte of channel 1 and A_0 is the least significant byte of A. B_x represents the x^{th} byte of channel 2 and B_0 is the least significant byte of B.

Figure 48-17. SPDIFTX_CDR Organization when SPDIFTX_MR.BPS=3



48.6.4 Channel Status and User Data Bits in SPDIF Frame

The SPDIFTX provides two methods to manage the channel status and user data bits transmitted in the SPDIF frame:

- The first method provides a direct mapping of the transmitted bits on a 192-bit configurable register. Each channel has its dedicated 192-bit register.
- The second method provides a way to configure the channel status and user data bits for each frame by writing their values together with the payload.

48.6.4.1 Direct Mapping Registers

The registers SPDIFTX_CHySx and SPDIFTX_CHyUDx configure the 192-bit Channel Status and User Data for each channel.

In order to use these registers to configure the 192-bit Channel Status and User Data for each channel, the following settings must be respected:

- SPDIFTX_MR.CMODE must be lower than ‘3’.
- SPDIFTX_EMR.CSM must not be set to ‘1’.
- SPDIFTX_EMR.UDM must not be set to ‘1’.

The SPDIFTX_CHySx and SPDIFTX_CHyUDx registers must be configured before enabling the SPDIFTX. Then, if the Channel Status or User Data value must be changed for each new 192-frame block, SPDIFTX_ISR.CSRDY and SPDIFTX_ISR.UDRDY flags indicate when the value of the CHySx and CHyUDx can be changed.

Note: The registers must be written consecutively to ensure the data is ready for the next block.

48.6.4.2 Frame-Oriented Configuration

If SPDIFTX_MR.CMODE=3, SPDIFTX_EMR.CSM is set to ‘1’ and SPDIFTX_EMR.UDM is set to ‘1, then the Channel Status bit and the User Data bit to send are supplied with the data written in SPDIFTX_CDR. The fields SPDIFTX_CDR.CS and SPDIFTX_CDR.UD indicate the value to send in the subframe.

48.6.5 Transmit FIFO

Each channel embeds a 32-word FIFO to handle the data to be transmitted (words are defined on 24 bits). The depth of the FIFO varies with the size of the audio words defined by SPDIFTX_MR.BPS. The size of the FIFO in bytes is 32 multiplied by three. The number of bytes that are written into the FIFO depends on the word size (SPDIFTX_MR.BPS) and on the transfer mode (SPDIFTX_MR.CMODE).

As long as the TXRDY flag of a channel in the SPDIF Transmitter Channel Status register is active, the SPDIFTX controller is ready to accept one data through SPDIFTX_CDR. Data which cannot be converted immediately are stored in the FIFO of the corresponding channel.

If the FIFO of one channel is full, the TXFULL flag rises and the TXRDY flag is inactive.

If the system writes data in SPDIFTX_CDR while the TXFULL flag is high, an overrun occurs. The data is not written in the FIFO and the TXOVR flag rises.

If the FIFO of one channel is empty, the TXEMPTY flag is raised.

If the system fails to send data while the TXEMPTY flag is raised, an underrun occurs. The last transmitted data is sent again and the TXUDR flag is raised.

A threshold can also be defined to know when the FIFO is ready to receive an amount of data defined by SPDIFTX_MR.CHUNK. The threshold is defined as a number of accesses to SPDIFTX_CDR.

This chunk definition can be used with the DMA to define how many access can be performed. In this case, the chunk size that is chosen must match the chunk value defined in the DMA.

48.6.6 Write Protection Registers

To prevent any single software error from corrupting SPDIFTX behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the SPDIFTX Write Protection Mode Register (SPDIFTX_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SPDIFTX Write Protection Status Register (SPDIFTX_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SPDIFTX_WPSR.

The following register can be write-protected by setting SPDIFTX_WPMR.WPEN:

- SPDIFTX Mode Register

The following registers can be write-protected by setting SPDIFTX_WPMR.WPITEN:

- SPDIFTX Interrupt Enable Register
- SPDIFTX Interrupt Disable Register

48.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SPDIFTX_CR	31:24								
		23:16								
		15:8								
		7:0							FCLR	SWRST
0x04	SPDIFTX_MR	31:24	DUDCPY	DCSCPY	BPS[1:0]		DNFR[1:0]		VALID2	VALID1
		23:16				CHUNK[4:0]				
		15:8			VBPS[5:0]					
		7:0			CMODE[1:0]		JUSTIFY	ENDIAN	MULTICH	TXEN
0x08	SPDIFTX_EMR	31:24								
		23:16								
		15:8								
		7:0				VALIDM	PARM	CSM	UDM	PCM
0x0C	SPDIFTX_CDR	31:24	CDR[31:24]							
		23:16	CDR[23:16]							
		15:8	CDR[15:8]							
		7:0	CDR[7:0]							
0x0C	SPDIFTX_CDR (CONTROL_BITS)	31:24			PC[1:0]		PAR	CS	UD	VALID
		23:16	CDR[23:16]							
		15:8	CDR[15:8]							
		7:0	CDR[7:0]							
0x10	SPDIFTX_SR	31:24								
		23:16								
		15:8								
		7:0								ENS
0x14	SPDIFTX_IER	31:24								
		23:16								
		15:8			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
		7:0	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
0x18	SPDIFTX_IDR	31:24								
		23:16								
		15:8			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
		7:0	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
0x1C	SPDIFTX_IMR	31:24								
		23:16								
		15:8			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
		7:0	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
0x20	SPDIFTX_ISR	31:24								
		23:16								
		15:8			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
		7:0	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
0x24	SPDIFTX_SFI	31:24								
		23:16								
		15:8							SFI[9:8]	
		7:0	SFI[7:0]							
0x28 ... 0x2F	Reserved									
0x30	SPDIFTX_AW1	31:24	AW[31:24]							
		23:16	AW[23:16]							
		15:8	AW[15:8]							
		7:0	AW[7:0]							
0x34	SPDIFTX_AW2	31:24	AW[31:24]							
		23:16	AW[23:16]							
		15:8	AW[15:8]							
		7:0	AW[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38 ... 0x4F	Reserved									
0x50	SPDIFTX_CH1UD0	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x54	SPDIFTX_CH1UD1	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x58	SPDIFTX_CH1UD2	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x5C	SPDIFTX_CH1UD3	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x60	SPDIFTX_CH1UD4	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x64	SPDIFTX_CH1UD5	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x68	SPDIFTX_CH2UD0	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x6C	SPDIFTX_CH2UD1	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x70	SPDIFTX_CH2UD2	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x74	SPDIFTX_CH2UD3	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x78	SPDIFTX_CH2UD4	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x7C	SPDIFTX_CH2UD5	31:24	CHUD[31:24]							
		23:16	CHUD[23:16]							
		15:8	CHUD[15:8]							
		7:0	CHUD[7:0]							
0x80	SPDIFTX_CH1S0	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x84	SPDIFTX_CH1S1	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x88	SPDIFTX_CH1S2	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x8C	SPDIFTX_CH1S3	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x90	SPDIFTX_CH1S4	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x94	SPDIFTX_CH1S5	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x98	SPDIFTX_CH2S0	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0x9C	SPDIFTX_CH2S1	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0xA0	SPDIFTX_CH2S2	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0xA4	SPDIFTX_CH2S3	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0xA8	SPDIFTX_CH2S4	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0xAC	SPDIFTX_CH2S5	31:24	CHS[31:24]							
		23:16	CHS[23:16]							
		15:8	CHS[15:8]							
		7:0	CHS[7:0]							
0xB0 ... 0xDF	Reserved									
0xE0	SPDIFTX_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE4	SPDIFTX_WPSR	31:24							SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE			WPVS

48.7.1 SPDIF Transmitter Control Register

Name: SPDIFTX_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							FCLR	SWRST
Access							W	W
Reset							–	–

Bit 1 – FCLR FIFO Clear

Value	Description
0	No effect.
1	Empties Channel 1 and Channel 2 FIFOs.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the SPDIFTX interface.

48.7.2 SPDIF Transmitter Mode Register

Name: SPDIFTX_MR
Offset: 0x04
Reset: 0x23011806
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPDIFTX Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DUDCPY	DCSCPY	BPS[1:0]		DNFR[1:0]		VALID2	VALID1
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	1	0	0		1	1

Bit	23	22	21	20	19	18	17	16
				CHUNK[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
			VBPS[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	1	1	0	0	0

Bit	7	6	5	4	3	2	1	0
			CMODE[1:0]		JUSTIFY	ENDIAN	MULTICH	TXEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	1	0

Bit 31 – DUDCPY Disable User Data Copy

Value	Description
0	Any data written in SPDIFTX_CH1UDx registers is copied to SPDIFTX_CH2UDx.
1	SPDIFTX_CH1UDx and SPDIFTX_CH2UDx are independent.

Bit 30 – DCSCPY Disable Channel Status Copy

Value	Description
0	Any data written in SPDIFTX_CH1Sx registers is copied to SPDIFTX_CH2Sx.
1	SPDIFTX_CH1Sx and SPDIFTX_CH2Sx are independent.

Bits 29:28 – BPS[1:0] Bytes Per Sample
This field defines the SPDIFTX_CDR operating mode by assigning a size to the data container holding each sample. See Data Organization.

Value	Name	Description
0	BYTE1	SPDIFTX_CDR data holding operating mode is optimized for data size up to 8 bits.
1	BYTE2	SPDIFTX_CDR data holding operating mode is optimized for data size in range 9 to 16 bits.
2	BYTE3	SPDIFTX_CDR data holding operating mode is optimized for data size in range 17 to 24 bits.
3	BYTE4	SPDIFTX_CDR data holding operating mode is optimized for data size in range 25 to 32 bits.

Bits 28:27 – DNFR[1:0] Disable Null Frame on Underrun

Value	Description
0	In case of underrun (SPDIFTX_ISR.TXEMPTY flag is set and SPDIFTX needs a data), invalid frames are sent (Validity bit at level '1').
1	In case of underrun (SPDIFTX_ISR.TXEMPTY flag is set and SPDIFTX needs a data), valid frames are sent.

Bit 25 – VALID2 Validity Bit Channel 2

Bit 24 – VALID1 Validity Bit Channel 1

Bits 20:16 – CHUNK[4:0] Chunk Size
Defines the size of the chunk. The minimum value of CHUNK is 1.
When the TX FIFO can accept CHUNK data transfer through SPDIFTX_CDR, the TXCHUNK flag is raised.

Bits 13:8 – VBPS[5:0] Valid Bits Per Sample
This field is used to explicitly indicate how many bits of precision are present in the signal.
VBPS must be less than or equal to the number of bits of the container defined in SPDIFTX_MR.BPS and must be greater than 1.
If VBPS is less than the number of bits per sample of the container defined by SPDIFTX_MR.BPS, then the data alignment is defined in SPDIFTX_MR.JUSTIFY.

Bits 5:4 – CMODE[1:0] Common Audio Register Transfer Mode (if BPS=2)
CMODE has no effect when SPDIFTX_MR.BPS is not equal to 2.

Value	Name	Description
0	CHANNEL_INDEX	SPDIFTX_CDR.CDR[25:24] indicates the channel on which the data CDR[23:0] will be transmitted (1 for channel 1, 2 for channel 2). Transmitted data are located on SPDIFTX_CDR.CDR[23:0].
1	CHANNELS_TOGGLING	The data are stored alternately in the FIFO of channel 1 and channel 2. The first sent data after a software or hardware reset is stored in the FIFO of channel 1. Transmitted data are located on SPDIFTX_CDR.CDR[23:0].
2	COMPACT_24BIT	This mode is optimized for 24-bit data compacted on a 32-bit memory space. Transmitted data are located on SPDIFTX_CDR.CDR[31:0]. The 32 bits of SPDIFTX_CDR are used and contain one or more channel data.
3	CONTROL_BITS	Control bits (preamble, validity, user data, channel status, parity) can be managed by writing SPDIFTX_CDR.CDR[31:24] depending on SPDIFTX_EMR configuration. The data are stored alternately in the FIFO of channel 1 and channel 2. The first sent data after a software or hardware reset is stored in the FIFO of channel 1. Transmitted data are located on SPDIFTX_CDR.CDR[23:0].

Bit 3 – JUSTIFY Data Justification

Value	Name	Description
0	LSB	Least Significant Bit justification. The valid bits of the signal are aligned on the least-significant bits of the container.
1	MSB	Most Significant Bit justification. The valid bits of the signal are aligned on the most-significant bits of the container.

Bit 2 – ENDIAN Data Word Endian Mode

Value	Name	Description
0	LITTLE	Little-endian mode.
1	BIG	Big-endian mode.

Bit 1 – MULTICH Multichannel Transfer

Value	Name	Description
0	MONO	One channel is sent on channel 1 to SPDIFTX_TX and is copied on channel 2.
1	DUAL	Two separate channels are sent to SPDIFTX_TX.

Bit 0 – TXEN SPDIFTX Transmit Enable

Value	Name	Description
0	DISABLE	SPDIFTX transmission is disabled.
1	ENABLE	SPDIFTX transmission is enabled.

48.7.3 SPDIF Transmitter Extended Mode Register

Name: SPDIFTX_EMR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in SPDIFTX_WPMR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				VALIDM	PARM	CSM	UDM	PCM
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – VALIDM Validity Bit Mode

Value	Description
0	Validity bit is defined by SPDIFTX_MR.VALID1 and SPDIFTX_MR.VALID2 values.
1	Validity bit is defined by SPDIFTX_CDR.VALID.

Bit 3 – PARM Parity Mode

Value	Description
0	Parity bit is automatically set by the SPDIFTX.
1	Parity bit sent is defined by SPDIFTX_CDR.PAR.

Bit 2 – CSM Channel Status Mode

Value	Description
0	Channel status is defined by SPDIFTX_CHySx.
1	Channel status is defined by SPDIFTX_CDR.CS.

Bit 1 – UDM User Data Mode

Value	Description
0	User data is defined by SPDIFTX_CHyUDx.
1	User data is defined by SPDIFTX_CDR.UD.

Bit 0 – PCM Preamble Code Mode

Value	Description
0	Preamble code is generated automatically by the SPDIFTX.
1	Preamble code is defined by SPDIFTX_CDR.PC.

48.7.4 SPDIF Transmitter Common Data Register

Name: SPDIFTX_CDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	CDR[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	CDR[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	CDR[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	CDR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – CDR[31:0] Common Data Register
Data sent to channel 1 and/or channel 2.
The mapping of the register depends on the transfer configuration defined in SPDIFTX_MR.

48.7.5 SPDIF Transmitter Common Data Register (CONTROL_BITS)

Name: SPDIFTX_CDR (CONTROL_BITS)
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
			PC[1:0]		PAR	CS	UD	VALID
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	CDR[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CDR[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CDR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 29:28 – PC[1:0] Preamble Code

Value	Name	Description
0	PREAMBLE_B	Preamble “B” is sent.
1	PREAMBLE_M	Preamble “M” is sent.
	PREAMBLE_W	Preamble “W” is sent.

Bit 27 – PAR Parity

Value	Description
0	Correct parity bit is sent.
1	Wrong parity bit is sent.

Bit 26 – CS Channel Status
Channel status bit to send.

Bit 25 – UD User Data
User bit to send.

Bit 24 – VALID Validity Bit

Value	Description
0	Sample is valid for analog conversion.
1	Sample is not valid for analog conversion.

Bits 23:0 – CDR[23:0] Common Data Register
Data sent to channel 1 and/or channel 2.
The mapping of the register depends on the transfer configuration defined in SPDIFTX_MR.

48.7.6 SPDIF Transmitter Status Register

Name: SPDIFTX_SR
Offset: 0x10
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENS
Access								W
Reset								–

Bit 0 – ENS Enable Status

Value	Description
0	SPDIF core is disabled.
1	SPDIF core is enabled.

48.7.7 SPDIF Transmitter Interrupt Enable Register

Name: SPDIFTX_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPDIFTX Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 13 – BEND Block End Interrupt Enable

Bit 12 – TXUDR2 Channel 2 Transmit Under Flow Interrupt Enable

Bit 11 – TXUDR1 Channel 1 Transmit Under Flow Interrupt Enable

Bit 10 – SECE Security Report Interrupt Enable

Bit 9 – TXRDYCH2 Transmit Ready Channel 2 Interrupt Enable

Bit 8 – TXRDYCH1 Transmit Ready Channel 1 Interrupt Enable

Bit 7 – UDRDY User Data Ready Interrupt Enable

Bit 6 – CSRDY Channel Status Ready Interrupt Enable

Bit 5 – TXOVR Transmit Over Flow Interrupt Enable

Bit 4 – TXUDR Transmit Under Flow Interrupt Enable

Bit 3 – TXCHUNK Transmit FIFO Chunk Size Empty Interrupt Enable

Bit 2 – TXFULL Transmit FIFO Full Interrupt Enable

Bit 1 – TXEMPTY Transmit FIFO Empty Interrupt Enable

Bit 0 – TXRDY Transmit Ready Interrupt Enable

48.7.8 SPDIF Transmitter Interrupt Disable Register

Name: SPDIFTX_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPDIFTX Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 13 – BEND Block End Interrupt Disable

Bit 12 – TXUDR2 Channel 2 Transmit Under Flow Interrupt Disable

Bit 11 – TXUDR1 Channel 1 Transmit Under Flow Interrupt Disable

Bit 10 – SECE Security Report Interrupt Disable

Bit 9 – TXRDYCH2 Transmit Ready Channel 2 Interrupt Disable

Bit 8 – TXRDYCH1 Transmit Ready Channel 1 Interrupt Disable

Bit 7 – UDRDY User Data Ready Interrupt Disable

Bit 6 – CSRDY Channel Status Ready Interrupt Disable

Bit 5 – TXOVR Transmit Over Flow Interrupt Disable

Bit 4 – TXUDR Transmit Under Flow Interrupt Disable

Bit 3 – TXCHUNK Transmit FIFO Chunk Size Empty Interrupt Disable

Bit 2 – TXFULL Transmit FIFO Full Interrupt Disable

Bit 1 – TXEMPTY Transmit FIFO Empty Interrupt Disable

Bit 0 – TXRDY Transmit Ready Interrupt Disable

48.7.9 SPDIF Transmitter Interrupt Mask Register

Name: SPDIFTX_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
Access			W	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 13 – BEND Block End Interrupt Mask

Bit 12 – TXUDR2 Channel 2 Transmit Under Flow Interrupt Mask

Bit 11 – TXUDR1 Channel 1 Transmit Under Flow Interrupt Mask

Bit 10 – SECE Security Report Interrupt Mask

Bit 9 – TXRDYCH2 Transmit Ready Channel 2 Interrupt Mask

Bit 8 – TXRDYCH1 Transmit Ready Channel 1 Interrupt Mask

Bit 7 – UDRDY User Data Ready Interrupt Mask

Bit 6 – CSRDY Channel Status Ready Interrupt Mask

Bit 5 – TXOVR Transmit Over Flow Interrupt Mask

Bit 4 – TXUDR Transmit Under Flow Interrupt Mask

Bit 3 – TXCHUNK Transmit FIFO Chunk Size Empty Interrupt Mask

Bit 2 – TXFULL Transmit FIFO Full Interrupt Mask

Bit 1 – TXEMPTY Transmit FIFO Empty Interrupt Mask

Bit 0 – TXRDY Transmit Ready Interrupt Mask

48.7.10 SPDIF Transmitter Interrupt Status Register

Name: SPDIFTX_ISR
Offset: 0x20
Reset: 0x000003CB
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			BEND	TXUDR2	TXUDR1	SECE	TXRDYCH2	TXRDYCH1
Access			R	R	R	R	R	R
Reset			0	0	0	0	1	1

Bit	7	6	5	4	3	2	1	0
	UDRDY	CSRDY	TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	0	1	0	1	1

Bit 13 – BEND Block End Status (cleared on read)

Value	Description
0	No Block End event occurred since the last read of SPDIFTX_ISR.
1	At least one Block End event occurred since the last read of SPDIFTX_ISR.

Bit 12 – TXUDR2 Channel 2 Transmit Underflow Status (cleared on read)

Value	Description
0	Channel 2 Transmit FIFO has been filled on time.
1	Channel 2 Transmit FIFO has not been filled on time.

Bit 11 – TXUDR1 Channel 1 Transmit Underflow Status (cleared on read)

Value	Description
0	Channel 1 Transmit FIFO has been filled on time.
1	Channel 1 Transmit FIFO has not been filled on time.

Bit 10 – SECE Security Report Status (cleared on read)

Value	Description
0	There is no security report in SPDIFTX_WPSR.
1	One security flag is set in SPDIFTX_WPSR.

Bit 9 – TXRDYCH2 Transmit Ready Status Channel 2 (cleared by writing SPDIFTX_CDR or SPDIFTX_AW2)

Value	Description
0	Channel 2 Transmit FIFO is full and cannot accept more data.
1	Channel 2 Transmit FIFO is not full; one data can be written to SPDIFTX_CDR or SPDIFTX_AW2.

Bit 8 – TXRDYCH1 Transmit Ready Status Channel 1 (cleared by writing SPDIFTX_CDR or SPDIFTX_AW1)

Value	Description
0	Channel 1 Transmit FIFO is full and cannot accept more data.

Value	Description
1	Channel 1 Transmit FIFO is not full; one data can be written to SPDIFTX_CDR or SPDIFTX_AW1.

Bit 7 – UDRDY User Data Ready Status (cleared writing SPDIFTX_CHyUDR0 register)

Value	Description
0	SPDIFTX_CHyUDx register cannot accept data.
1	A data can be written to SPDIFTX_CHyUDx.

Bit 6 – CSRDY Channel Status Ready Status (cleared writing SPDIFTX_CHyS0 register)

Value	Description
0	SPDIFTX_CHySx register cannot accept data.
1	A data can be written to SPDIFTX_CHySx.

Bit 5 – TXOVR Transmit Over Flow Status (cleared on read)

Value	Description
0	No Transmit FIFO overflow occurred since last read of SPDIFTX_ISR.
1	A data has been written while the transmit FIFO was full.

Bit 4 – TXUDR Transmit Under Flow Status (cleared on read)

Value	Description
0	All transmit FIFOs has been filled on time.
1	One of the Transmit FIFOs has not been filled on time.

Bit 3 – TXCHUNK Transmit FIFO Chunk Size Empty Status (cleared by writing SPDIFTX_CDR, SPDIFTX_AW1 or SPDIFTX_AW2 register)

Value	Description
0	The TX FIFOs cannot accept SPDIFTX_MR.CHUNK data transfer through SPDIFTX_CDR.
1	The TX FIFOs can accept SPDIFTX_MR.CHUNK data transfer through SPDIFTX_CDR.

Bit 2 – TXFULL Transmit FIFO Full Status (cleared when datas are sent or writing the SPDIFTX_CR.FCLR bit)

Value	Description
0	Transmit FIFOs are not full.
1	Transmit FIFOs are full.

Bit 1 – TXEMPTY Transmit FIFO Empty Status (cleared by writing SPDIFTX_CDR, SPDIFTX_AW1 or SPDIFTX_AW2 register)

Value	Description
0	Transmit FIFOs are not empty.
1	Transmit FIFOs are empty.

Bit 0 – TXRDY Transmit Ready Status (cleared by writing SPDIFTX_CDR, SPDIFTX_AW1 or SPDIFTX_AW2 register)

Value	Description
0	Transmit FIFOs are full and cannot accept more data
1	Transmit FIFOs are not full; one data can be written to SPDIFTX_CDR

48.7.11 SPDIF Transmitter Subframe Index

Name: SPDIFTX_SFI
Offset: 0x24
Reset: 0x0000017F
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							SFI[9:8]	
Access							R	R
Reset							0	1

Bit	7	6	5	4	3	2	1	0
	SFI[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	1	1	1	1	1

Bits 9:0 – SFI[9:0] Transmit Subframe Index
Index of the currently sent subframe.
Even values correspond to channel 1. Odd values correspond to channel 2.
As specified by the IEC-60958 standard, there are 192 × 2 subframes.
Due to the asynchronous transfer, a minimum of two and a maximum of three accesses must be performed to know the currently sent subframe index. When two successive reads of this register give the same value, the data is valid.

48.7.12 SPDIF Transmitter Channel 1 Audio Word Register

Name: SPDIFTX_AW1
Offset: 0x30
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	AW[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AW[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AW[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AW[31:0] Audio Word Channel 1
Data sent to channel 1.

48.7.13 SPDIF Transmitter Channel 2 Audio Word Register

Name: SPDIFTX_AW2
Offset: 0x34
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	AW[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AW[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AW[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AW[31:0] Audio Word Channel 2
Data sent to channel 2.

48.7.14 SPDIF Transmitter Channel 1 User Data Register x

Name: SPDIFTX_CH1UDx
Offset: 0x50 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHUD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHUD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHUD[31:0] Channel 1 User Data Word x
The six 32-bit User Data registers contain the 192-bit User Data sent to channel 1.

48.7.15 SPDIF Transmitter Channel 2 User Data Register x

Name: SPDIFTX_CH2UDx
Offset: 0x68 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHUD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHUD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHUD[31:0] Channel 2 User Data Word x
The six 32-bit User Data registers contain the 192-bit User Data sent to channel 2.

48.7.16 SPDIF Transmitter Channel 1 Status Register x

Name: SPDIFTX_CH1Sx
Offset: 0x80 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHS[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHS[31:0] Channel 1 Status Word x
The six 32-bit Channel Status registers contain the 192-bit Channel Status sent to channel 1.

48.7.17 SPDIF Transmitter Channel 2 Status Register x

Name: SPDIFTX_CH2Sx
Offset: 0x98 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHS[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHS[31:0] Channel 2 Status Word x
The six 32-bit Channel Status registers contain the 192-bit Channel Status sent to channel 2.

48.7.18 SPDIF Transmitter Write Protection Mode Register

Name: SPDIFTX_WPMR
Offset: 0xE0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x444946	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN. Always reads at 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in SPDIFTX_WPSR.WPSRC and the last software control error type is reported in SPDIFTX_WPSR.SWETYP. The SPDIFTX_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in SPDIFTX_WPSR.WPSRC and only the first software control error type is reported in SPDIFTX_WPSR.SWETYP. The SPDIFTX_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register if WPKEY corresponds to0x444946.
1	Enables the write protection on the Control register if WPKEY corresponds to 0x444946.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to0x444946.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x444946.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x444946 (“DIF” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x444946 (“DIF” in ASCII).

48.7.19 SPDIF Transmitter Write Protection Status Register

Name: SPDIFTX_WPSR
Offset: 0xE4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							SWETYP[1:0]	
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE			WPVS
Access					R			R
Reset					0			0

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	UNDEF_RW	Access to an undefined address (warning).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SPDIFTX_WPSR.
1	A software error has occurred since the last read of SPDIFTX_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPSRC (if WPVS=0).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of SPDIFTX_WPSR.
1	A write protection violation has occurred since the last read of SPDIFTX_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

49. Pulse Density Microphone Controller (PDMC)

49.1 Description

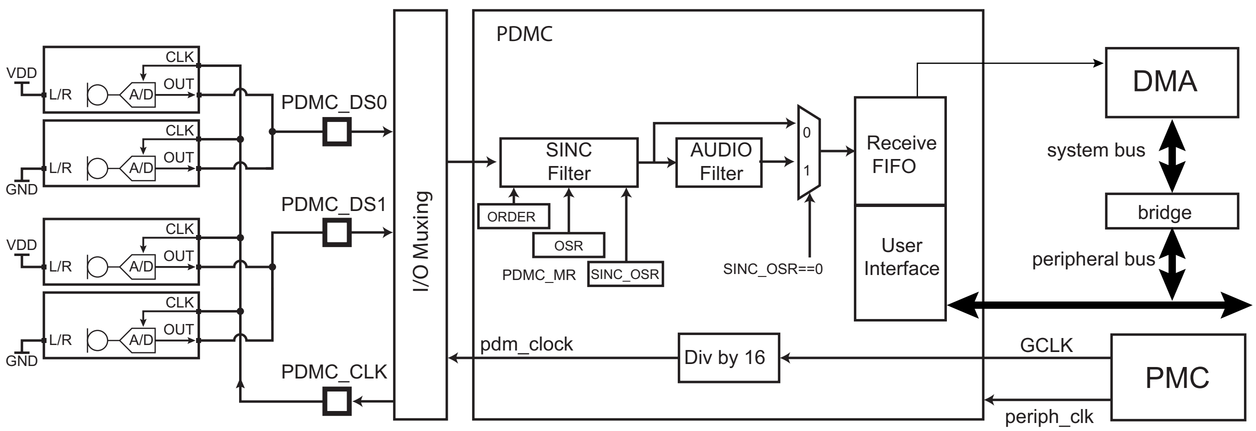
The Pulse Density Microphone Controller (PDMC) interfaces up to 4 digital microphones having Pulse Density Modulated (PDM) outputs. It generates a single clock line and samples 1 or 2 data lines. The signal path includes an audio grade programmable decimation filter and outputs 24-bit audio words on the APB bus.

49.2 Embedded Characteristics

- Up to 4 Microphone Channels
- Programmable OSR SINC Filter
- Programmable OSR Audio Filter
- Trigger Signals for DMA Channel
- PDM Microphone Clock Generation

49.3 PDMC Block Diagram

Figure 49-1. Block Diagram



49.4 Signal Description

Table 49-1. Pin Description

Pin Name	Description	Direction
PDMC_DS0	Data input for 2 PDM microphones	Input
PDMC_DS1	Data input for 2 PDM microphones	Input
PDMC_CLK	Clock output for PDM microphones	Output

49.5 Product Dependencies

49.5.1 Power Management

The PDMC can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the PDMC clocks.

49.5.2 Interrupt Sources

The PDMC interface has an interrupt line connected to the interrupt controller. Handling the PDMC interrupt requires programming the interrupt controller before configuring the PDMC.

49.6 Functional Description

The PDMC generates a clock for driving up to 4 PDM microphones. It embeds a SINC filter with programmable order (1 to 5) and an audio filter.

The audio filter can be disabled if specific filtering is required and performed by software.

The overall oversampling rate (OSR) can be configured to 64, 128 or 256 when the audio filter is enabled.

When the audio filter is disabled, the OSR of the SINC filter can be configured to 8, 16, 32, 64, 128 or 256.

49.6.1 Clocks and Output Sampling Rate

The clock sent to the PDM microphone is GCLK/16.

The controller continuously decimates the incoming streams. The output sampling rate depends on PDMC_MR.OSR).

Table 49-2. Conversion Rates

PDMC_MR.SINC_OSR	PDMC_MR.OSR	Audio Filter	OSR[1]	Description and Oversampling Ratio (OSR[1])
0	0	Yes	–	Forbidden
	1	Yes	64	Audio filtering with OSR[1] = $2^{(PDMC\_MR.OSR+5)}$
	2	Yes	128	
	3	Yes	256	
> 0	–	No	OSR[1] = $2^{(PDMC\_MR.SINC\_OSR+2)}$	Maximum SINC_OSR corresponds to an OSR[1] of 256

Note:

1. The OSR is defined by the ratio between the clock frequency of the PDM (i.e., GCLK/16) and the output sampling rate after filtering and decimation.

49.6.2 SINC Filter

The SINC filter implemented in the PDMC has a programmable order (1 to 5) and OSR (8, 16, 32, 64, 128 and 256).

The SINC filter oversampling rate cannot be configured if the audio filter is enabled (PDMC_MR.SINC_OSR=0) and thus OSR results only from the configuration of PDMC_MR.OSR.

The SINC filter is configured by the following fields:

- PDMC_MR.SINCORDER defines the order of the SINC filter from 1 to 5.
- PDMC_MR.SINC_OSR defines the OSR of the SINC filter (when SINC_OSR is greater than 0).
- If PDMC_MR.SINC_OSR=0, the audio filter is enabled and the OSR of the SINC filter cannot be configured. Thus the overall (SINC + audio filter) oversampling rate is configured in PDMC_MR.OSR.

49.6.3 Audio Filter

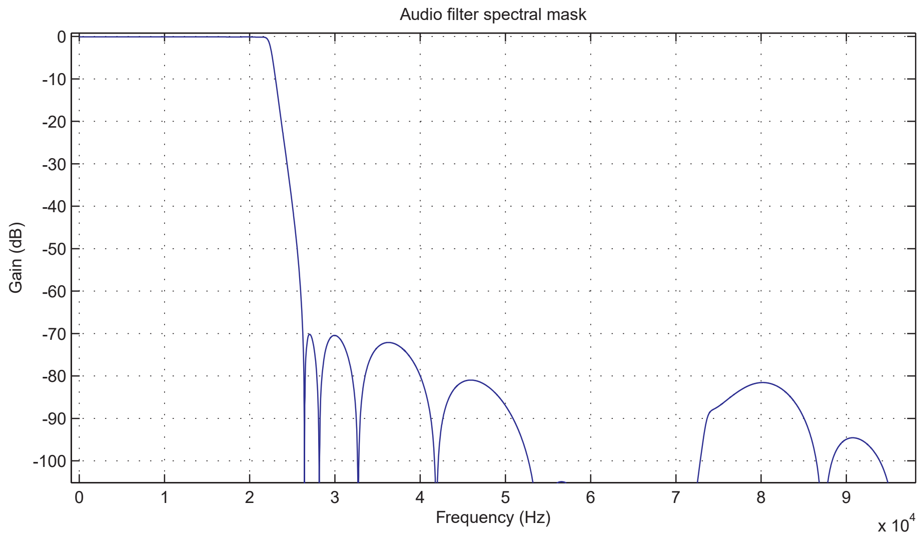
The PDMC embeds an audio filter which is combined to the SINC filter in order to optimize an audio signal. This audio filter is made of IIR filters and includes a droop compensation filter made of a third-order SINC filter.

The audio filter has a decimation factor of 4. The overall OSR is defined by the PDMC_MR.OSR field and is 64, 128 or 256.

When combined with the GCLK division factor of the PMC, the cut-off frequency can be optimized for any standard audio frequencies.

The figure below shows the spectral mask of the audio filter when the PDMC output frequency is configured to be 48 kHz. The mask scales with the output frequency.

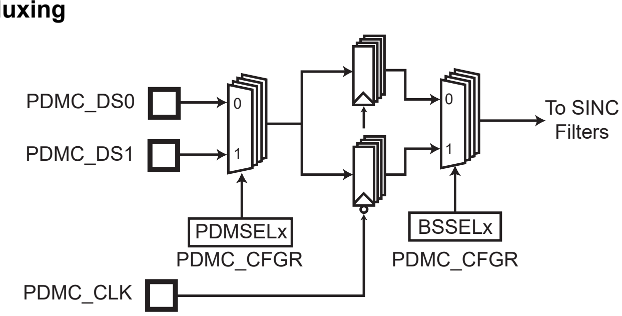
Figure 49-2. SINC and Audio Filter Spectral Mask



49.6.4 Input Muxing

The PDMC features four conversion channels. The source of each channel can be independently defined as PDMC_DS0 or PDMC_DS1 sampled at the rising or falling edge of PDMC_CLK by configuring PDMSELx and BSSELx in the Configuration register (PDMC_CFGR).

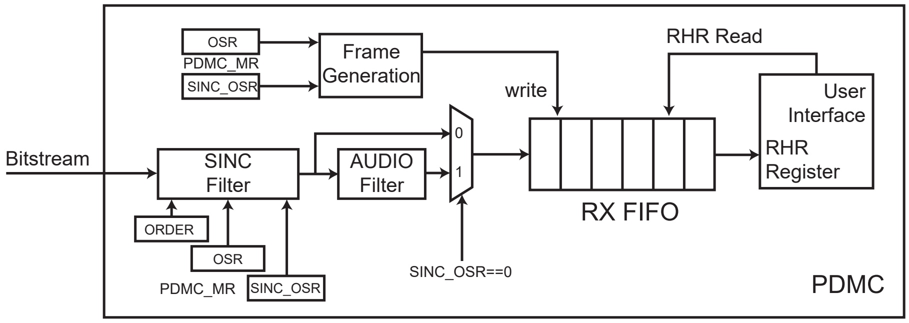
Figure 49-3. Input Muxing



49.6.5 Receive Channel

Converted data are first written into a FIFO, then read in the Receive Holding register (PDMC_RHR) register. Active channel conversion results are sent to PDMC_RHR in a cyclic way. This means that the enabled channel with the lowest index is sent first, followed by the other enabled channels, increasing the index at each step. Once the enabled channel with the highest index is sent, the next conversion result of the enabled channel with the lowest index is sent, etc.

Figure 49-4. Receive Channel Block Diagram



When at least one data is ready in the RX FIFO, the RXRDY flag rises and remains high as long as there is at least one data to be read in the FIFO.

When the entire FIFO has been filled with data, the RXFULL flag rises. If a new data is written into the RX FIFO while RXFULL is high, then the RXOVR flag rises. This flag remains high until the Interrupt Status register (PDMC_ISR) register is read.

Once all data contained in the RX FIFO are read, the RXEMPTY flag rises. If a data is read while the RX FIFO is empty, an underrun occurs and the RXUDR flag rises. This flag remains high until the PDMC_ISR register is read.

Once a data is written in the RX FIFO, the RXRDY flag rises. If the corresponding interrupt has been enabled, an interrupt is generated and remains high as long as a data is available in the RX FIFO.

The receive FIFO circuitry features DMA chunk size management. The PDMC can be configured to optimize the number of trigger events sent to the DMA.

When the number of audio data samples equals the chunk size configured in PDMC_MR.CHUNK, the RXCHUNK flag rises and a trigger event is sent to DMA. The flag is reset once CHUNK data are read.

49.6.6 Register Write Protection

To prevent any single software error from corrupting PDMC behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the PDMC Write Protection Mode Register (PDMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PDMC Write Protection Status Register (PDMC_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PDMC_WPSR.

The following registers can be write-protected by setting PDMC_WPMR.WPEN:

PDMC Mode Register

PDMC Configuration Register

The following registers can be write-protected by setting PDMC_WPMR.WPITEN:

PDMC Interrupt Enable Register

PDMC Interrupt Disable Register

49.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PDMC_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	PDMC_MR	31:24	CHUNK[3:0]				SINC_OSR[3:0]			
		23:16	SINCORDER[3:0]						OSR[1:0]	
		15:8								
		7:0					PDMCEN3	PDMCEN2	PDMCEN1	PDMCEN0
0x08	PDMC_CFGR	31:24								
		23:16		PDMSEL3		PDMSEL2		PDMSEL1		PDMSEL0
		15:8								
		7:0		BSSEL3		BSSEL2		BSSEL1		BSSEL0
0x0C	PDMC_RHR	31:24	CHANNEL[7:0]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x10 ... 0x13	Reserved									
0x14	PDMC_IER	31:24				WPERR				
		23:16								
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x18	PDMC_IDR	31:24				WPERR				
		23:16								
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x1C	PDMC_IMR	31:24				WPERR				
		23:16								
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x20	PDMC_ISR	31:24				WPERR				
		23:16						SECE		
		15:8								
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x24 ... 0x2B	Reserved									
0x2C	PDMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x30	PDMC_WPSR	31:24								
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0						SEQE		WPVS

49.7.1 PDMC Control Register

Name: PDMC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the PDMC interface.

49.7.2 PDMC Mode Register

Name: PDMC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PDMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CHUNK[3:0]				SINC_OSR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SINCORDER[3:0]						OSR[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PDMCEN3	PDMCEN2	PDMCEN1	PDMCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:28 – CHUNK[3:0] Chunk Size
Defines the DMA chunk size for RX channels.
Chunk size optimizes the number of trigger events sent to DMA for audio data transfer. Allowed values are 1, 2, 4, 8.
Refer to section "DMA Controller (DMAC)" for possible chunk sizes.
When the number of received data equals the value configured in PDMC_MR.CHUNK, the RXCHUNK flag rises.

Bits 27:24 – SINC_OSR[3:0] SINC Filter Oversampling Ratio

Value	Name	Description
0	DISABLE	Audio filtering mode. The SINC filter OSR is implicitly defined by the PDMC_MR.OSR field.
1	OSR8	The SINC filter OSR is 8.
2	OSR16	The SINC filter OSR is 16.
3	OSR32	The SINC filter OSR is 32.
4	OSR64	The SINC filter OSR is 64.
5	OSR128	The SINC filter OSR is 128.
6	OSR256	The SINC filter OSR is 256.

Bits 23:20 – SINCORDER[3:0] SINC Filter Order
Defines the SINC filter order.
Values outside those defined in the following table are forbidden.
SINCORDER must be set to 3 when in Audio Filtering mode (SINC_OSR=0), as the audio filtering compensates the droop of the third order SINC filter.

Value	Name	Description
1	ORDER1	SINC filter order is 1.
2	ORDER2	SINC filter order is 2.
3	ORDER3	SINC filter order is 3. Recommended if SINC_OSR=0
4	ORDER4	SINC filter order is 4.

Value	Name	Description
5	ORDER5	SINC filter order is 5.

Bits 17:16 – OSR[1:0] Audio Oversampling Ratio
Oversampling ratio between the frequency of the clock sent to the PDMC and the output sampling rate when SINC_OSR=0. Audio filtering is performed in addition to SINC filtering.

Value	Name	Description
0	–	Reserved
1	OSR64	OSR is 64.
2	OSR128	OSR is 128.
3	OSR256	OSR is 256.

Bits 0, 1, 2, 3 – PDMCENx PDMC Channel x Enable

Value	Name	Description
0	DISABLE	PDMC is disabled.
1	ENABLE	PDMC is enabled.

49.7.3 PDMC Configuration Register

Name: PDMC_CFGR
Offset: 0x08
Reset: 0x00500044
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PDMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		PDMSEL3		PDMSEL2		PDMSEL1		PDMSEL0
Access		R/W		R/W		R/W		R/W
Reset		1		1		0		0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		BSSEL3		BSSEL2		BSSEL1		BSSEL0
Access		R/W		R/W		R/W		R/W
Reset		1		0		1		0

Bits 16, 18, 20, 22 – PDMSELx PDM Microphone Source Selection

Value	Name	Description
0	DS0	PDMSELx corresponds to PMDC_DS0.
1	DS1	PDMSELx corresponds to PMDC_DS1.

Bits 0, 2, 4, 6 – BSSELx Bitstream Source Selection

Value	Description
0	The selected PDMC_DSx source is sampled on the positive edge of PDMC_CLK.
1	The selected PDMC_DSx source is sampled on the negative edge of PDMC_CLK.

49.7.4 PDMC Receive Holding Register

Name: PDMC_RHR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CHANNEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – CHANNEL[7:0] Channel Index of the Data
Index of the currently read data.

Bits 23:0 – DATA[23:0] Converted Data
This register contains the filtered data.

49.7.5 PDMC Interrupt Enable Register

Name: PDMC_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PDMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				WPERR				
Access				W				
Reset				–				
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 28 – WPERR Write Protection Error Enable

Bit 5 – RXOVR Receive Over Flow Interrupt Enable

Bit 4 – RXUDR Receive Under Flow Interrupt Enable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Enable

Bit 2 – RXFULL Receive FIFO Full Interrupt Enable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Enable

Bit 0 – RXRDY Receive Ready Interrupt Enable

49.7.6 PDMC Interrupt Disable Register

Name: PDMC_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PDMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				WPERR				
Access				W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 28 – WPERR Write Protection Error Disable

Bit 5 – RXOVR Receive Over Flow Interrupt Disable

Bit 4 – RXUDR Receive Under Flow Interrupt Disable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Disable

Bit 2 – RXFULL Receive FIFO Full Interrupt Disable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Disable

Bit 0 – RXRDY Receive Ready Interrupt Disable

49.7.7 PDMC Interrupt Mask Register

Name: PDMC_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
				WPERR				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 28 – WPERR Write Protection Error Mask

Bit 5 – RXOVR Receive Over Flow Interrupt Mask

Bit 4 – RXUDR Receive Under Flow Interrupt Mask

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Mask

Bit 2 – RXFULL Receive FIFO Full Interrupt Mask

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Mask

Bit 0 – RXRDY Receive Ready Interrupt Mask

49.7.8 PDMC Interrupt Status Register

Name: PDMC_ISR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				WPERR				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
						SECE		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 28 – WPERR Write Protection Error Status (cleared on read)

Value	Description
0	No write protection error event occurred since the last read of PDMC_ISR.
1	At least one write protection error event occurred since the last read of PDMC_ISR.

Bit 18 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event has occurred since the last read of PDMC_ISR.
1	One or more safety or security event occurred since the last read of PDMC_ISR. For details on the event, see PDMC Write Protection Status Register.

Bit 5 – RXOVR Receive Over Flow Interrupt Status (cleared on read)

Value	Description
0	No overflow event occurred since the last read of PDMC_ISR.
1	At least one overflow event occurred since the last read of PDMC_ISR.

Bit 4 – RXUDR Receive Under Flow Interrupt Status (cleared on read)

Value	Description
0	No underflow event occurred since the last read of PDMC_ISR.
1	At least one underflow event occurred since the last read of PDMC_ISR.

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Status

Value	Description
0	There is less than PDMC_MR.CHUNK data in the RX FIFO.
1	At least PDMC_MR.CHUNK data can be read in the RX FIFO.

Bit 2 – RXFULL Receive FIFO Full Interrupt Status

Value	Description
0	The RX FIFO is not full and can still receive data.

Value	Description
1	The RX FIFO is full and cannot receive more data.

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Status

Value	Description
0	At least one data is in the RX FIFO.
1	The RX FIFO is empty.

Bit 0 – RXRDY Receive Ready Interrupt Status

Value	Description
0	There is no data in the RX FIFO.
1	At least one data is in the RX FIFO and can be read through the PDMC_RHR register.

49.7.9 PDMC Write Protection Mode Register

Name: PDMC_WPMR
Offset: 0x2C
Reset: 0
Property: Read/Write

See 49.6.6. Register Write Protection for the list of write-protected registers.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
`0x50444D`	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always read at 0.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
`0`	Disables the write protection on interrupt registers if WPKEY corresponds to 0x50444D (“PDM” in ASCII).
`1`	Enables the write protection on interrupt registers if WPKEY corresponds to 0x50444D (“PDM” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
`0`	Disables write protection if WPKEY corresponds to 0x50444D (“PDM” in ASCII).
`1`	Enables write protection if WPKEY corresponds to 0x50444D (“PDM” in ASCII).

49.7.10 PDMC Write Protection Status Register

Name: PDMC_WPSR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

See 49.6.6. Register Write Protection for the list of write-protected registers.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						SEQE		WPVS
Access						R		R
Reset						0		0

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 2 – SEQE Internal Sequencer Error

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of PDMC_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of PDMC_WPSR.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of PDMC_WPSR.
1	A write protection violation has occurred since the last read of PDMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

50. Asynchronous Sample Rate Converter (ASRC)

50.1 Embedded Characteristics

- Up to 192 kHz Stereo Asynchronous Sample Rate Converter
- 8/10/12/14/16/18/20/24/32-bit Data Input/Output
- 24-bit Digital Signal Processing
- 115 dB Total Harmonic Distorsion+Noise
- Up to 4 Independent Stereo Channels
- Up to 8 DMA Channels
- Time-Division Multiplexing (TDM) Audio Streams Support
- Merge Audio Streams of Different Rates into a Single Stream
- Split an Audio Stream in Several Streams of Different Rates
- Channels Add/Drop and/or Split/Merge Capability
- Functional Safety Monitors and Reports
  - Register write protection
  - Detection of abnormal state in FIR filter controller

50.2 Description

The Asynchronous Sample Rate Converter (ASRC) converts the sample rate of an incoming audio frame without affecting quality. It supports input and output sampling rates up to 192 kHz.

The ASRC must be used when the rate of an input (or output) audio stream is different from the rate of the audio system processing the audio stream (for example, a 44.1 ksps audio stream must be processed in a 48 ksps environment).

The ASRC must be also used when the received (or transmitted) audio stream and the audio system both work at the same nominal rate, but are driven by different clock sources (two different crystal oscillators of same nominal clock frequency differing by a few ppm, for example).

ASRC is made of 4 independent digital signal processing (DSP) modules. Each DSP can be configured with specific input and output sampling frequencies and is associated to one of the DMA channels.

Each DSP can process mono or stereo audio streams.

Data can be provided to the DSPs via up to four input channels and data can be read from the DSPs via up to four output channels. A channel comprises a holding register that can be accessed via software or via a channel of the central DMA. The ASRC generates the trigger events for the central DMA.

Up to 8 channels are available.

Depending on the channel index and its configuration, a channel manages from one up to eight audio frames.

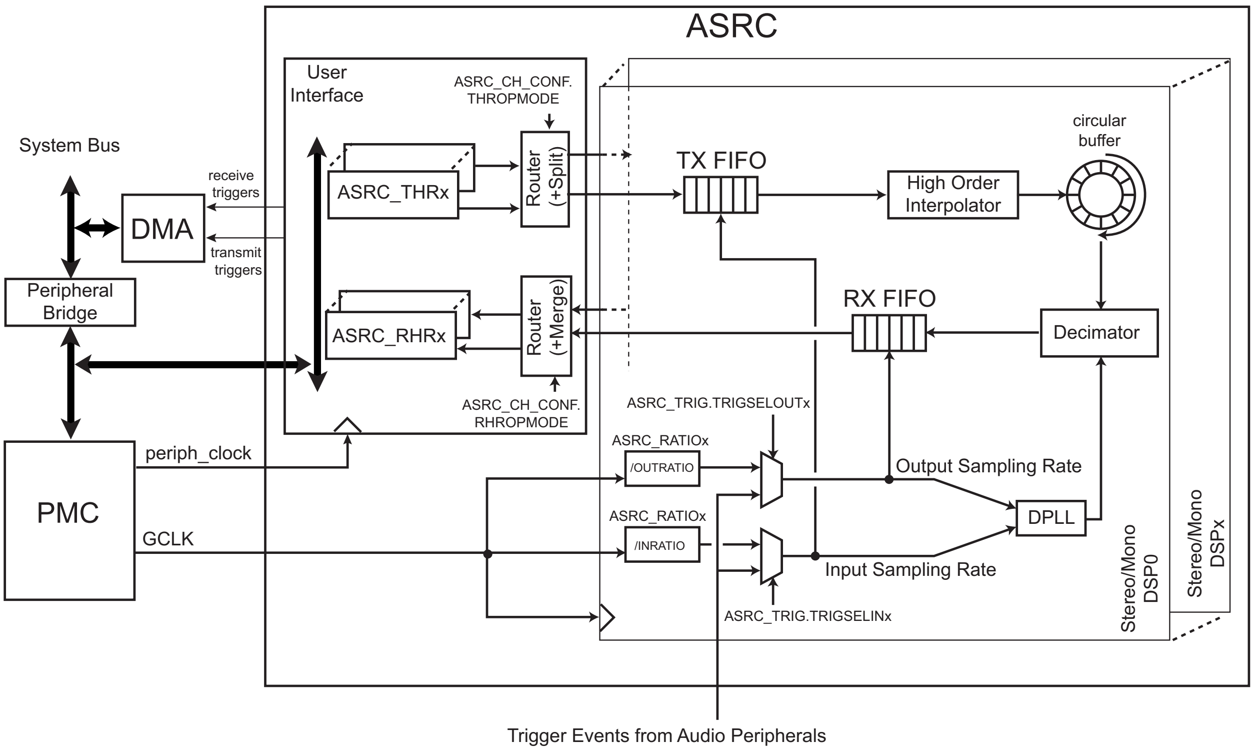
The ASRC can be configured to process up to four stereo audio streams in a fully independent manner. Input/out sampling frequencies differ for each DSP.

The ASRC supports TDM audio stream. ASRC is capable to concatenate/split audio streams while resampling frequencies. The ASRC can be configured to resample and merge several audio input sources of the same or of different sampling frequencies carried by several DMA channels into a single output audio stream sampled at a unique frequency and carried on a single DMA channel (TDM audio stream creation from multiple sources). The ASRC can also split and resample audio streams (TDM stream, for example) carried on a single DMA channel to multiple audio streams, each having the same or different resampling frequencies.

Note: There is no anti-aliasing filter. There is no limitation when up-sampling an audio stream but when down-sampling, the input signal frequency bandwidth upper boundary must be lower than half the resampling frequency.

50.3 Block Diagram

Figure 50-1. ASRC Block Diagram



50.4 Product Dependencies

50.4.1 Power Management

The ASRC clocks must be enabled prior to processing audio streams. The peripheral clock and GCLK clocks must be enabled in the Power Management Controller (PMC) before using the ASRC.

The peripheral clock is used by the interface to handle data and configuration sent to the interface of the controller, while the GCLK clock is used to clock the DSP sub-module.

50.4.2 Interrupt Sources

The ASRC interrupt line is connected on one of the internal sources of the interrupt controller. Using the ASRC interrupt requires the interrupt controller to be programmed first.

50.5 Functional Description

50.5.1 Preamble

Resampling an audio signal without using the ASRC affects the quality of the original audio signal.

The ASRC re-sampling quality is up to 115 dB THD+N without CPU intervention.

50.5.2 Limitations

The frequency ratio GCLK/peripheral clock must be lower than 4.

The frequency of the GCLK clock is determined by the maximum value of the input and output sampling frequencies.

When the sampling frequency is greater than 96 kHz (176.4 kHz, 192 kHz, for example), a ‘1’ must be written in ASRC_MR.GT96K and the maximum number of DSPs running in parallel is limited to 2. The GCLK frequency must be greater than or equal to 1024*maximum sampling frequency. For example, for a sampling frequency of 192 kHz, GCLK ≥ 1024*192 kHz

If three or four DSPs are required, then ASRC_MR.GT96K must be cleared, the maximum sampling frequency is limited to 96 kHz and the GCLK frequency must be greater than or equal to 2048*maximum sampling frequency. For example, for a sampling frequency of 48 kHz, GCLK ≥ 2048*48 kHz.

If up to two DSPs are required and the sampling frequency is up to 96 kHz, writing a ‘1’ in ASRC_MR.GT96K=1 leads to a lower frequency on GCLK. For example, for a sampling frequency of 48 kHz, GCLK ≥ 1024*48 kHz.

When up-sampling, the maximum ratio between output and input sampling frequencies is 16. When down-sampling, the max ratio is 12.

50.5.3 Sampling Frequencies and Oversampling

The ASRC converts an input audio stream with a sampling frequency defined either internally or externally to an audio stream at a configurable output sampling frequency generated either internally or externally.

Internal input or output sampling frequency is generated by a programmable division of GCLK. The division for input sampling frequency is defined in ASRC_MRx.INRATIO; the division for output sampling frequency is defined in ASRC_MRx.OUTRATIO.

The selection of the external trigger event for the input sampling frequency is made by configuring ASRC_TRIG.TRIGSELINx; the selection of the external trigger event for the output sampling frequency is made by configuring ASRC_TRIG.TRIGSELOUTx

If the input and output ratio are integer multiples, the ASRC behaves as a “synchronous” sample rate conversion.

The ASRC oversamples the incoming signal and then decimates the oversampled signal at the output sampling rate. The total oversampling ratio of the ASRC is 32768.

The input and output audio frequencies do not need to be synchronous.

Input and output sampling rates are defined by a division of the GCLK or by one of the signals in the table below.

Table 50-1. Trigger List

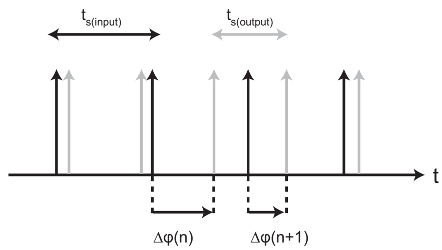
Trigger Index	Trigger Name
0	Internal trigger, a configurable division of GCLK
1	I2SMCC0 ws
2	I2SMCC1 ws
3	PDMC0 data synchro
4	PDMC1 data synchro
5	SSC0 RF
6	SSC0 TF
7	SSC1 RF
8	SSC1 TF
9	SPDIF TX
10	SPDIF RX
11	–
12	–
13	–
14	–

...........continued	
Trigger Index	**Trigger Name**
15	–

50.5.4 Embedded Digital PLL

Each DSP embeds a digital PLL (DPLL) for precise estimation of the sampling points.

Figure 50-2. Sampling Points



The embedded DPLL precisely computes the ratio between the input and the output sampling rates to obtain a high-quality output signal.

The DSP must be disabled (ASRC_MR.ASRCENx=0) if both input and output sampling rates are not established, as the DPLL continuously computes the sampling rate difference between the input and the output audio streams.

50.5.5 DSP and Channel Configuration

The ASRC comprises 4 independent 24-bit DSPs.

Each DSP can be configured to receive/convert/transmit either mono or stereo audio streams.

Each DSP features:

- an input and output sampling frequency defined in the Trigger Selection register (ASRC_TRIG)
- an input and output valid bit per sample defined in registers Valid bit Per Sample In (ASRC_VBPS_IN) and Valid bit Per Sample Out (ASRC_VBPS_OUT)
- transmit and receive FIFOs

Each DSP generates sources of interrupts that can be independently configured with the corresponding Interrupt registers (ASRC_IERx/ASRC_IDRx/ASRC_IMRx/ASRC_ISRx).

The DSP data is written through ASRC_THRx (transmit channels) and the DSP results are read from ASRC_RHRx (receive channels). The number of transfer channels (receive or transmit) equals the number of DSPs.

Each transfer channel can be configured to access one or more DSP.

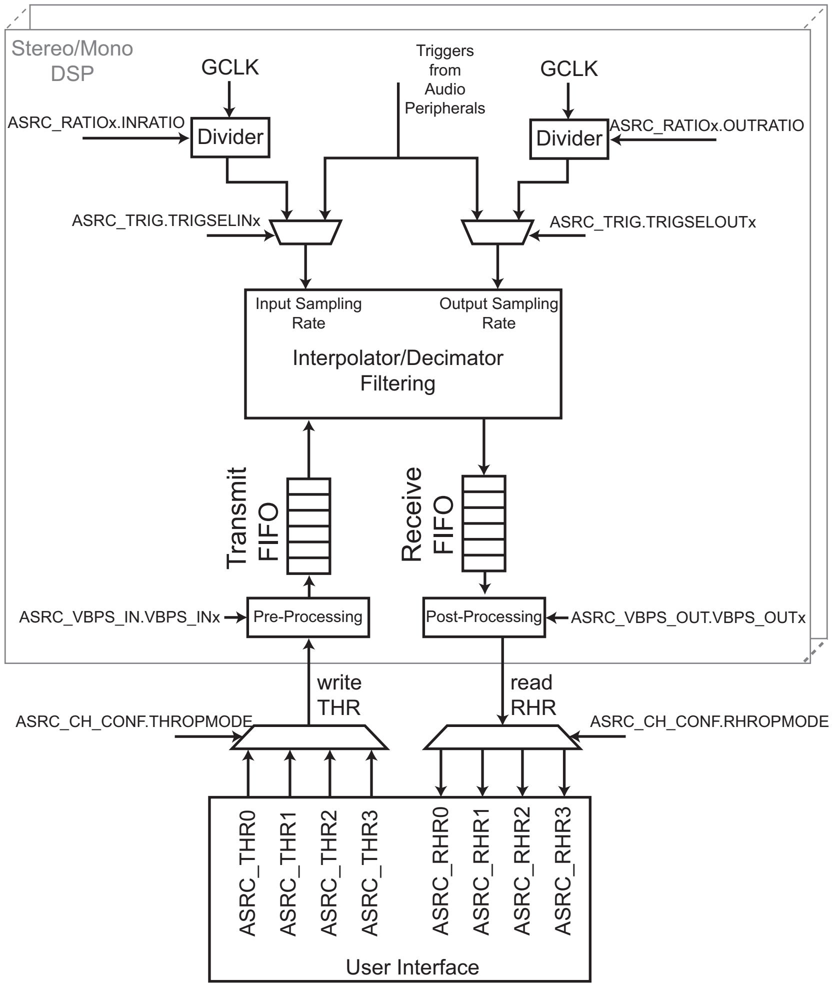
This configurable function allows concatenation of multiple audio streams transferred with different transmit channels into a single audio stream transferred on a single receive channel. It also allows the split of a single audio stream transferred with a transmit channel into several separate audio streams transferred on several receive channels.

The transfer channel management generates internal trigger events to interface with DMA controller.

The source of interrupts from the transfer channel management are mapped in the status registers together with the DSP sources of interrupt.

The ASRC generates a single interrupt line that may be driven by all DSP status registers.

Figure 50-3. DSP Datapath



50.5.5.1 Transmit Channel

Data are fed to the 32-word depth transmit FIFO of the DSP by writing into the ASRC_THRx that has been configured in the Channel Configuration register (ASRC_CH_CONF).

A data written in the ASRC_THRx register is automatically written into the corresponding transmit FIFO as long as the FIFO is ready to receive data (ASRC_ISRx.TXRDY=1).

When no more space is available in the transmit FIFO, the TXRDY flag falls and the TXFULL flag rises. Writing a data in the transmit FIFO when ASRC_ISRx.TXFULL =1 leads to an overflow and raises the ASRC_ISRx.TXOVR flag. The TXOVR flag is cleared on read.

The transmit FIFO is read at the sample rate defined in ASRC_TRIG.TRIGSELINx.

The DSP core reads the data transmit FIFO at the selected sampling frame rate. When the transmit FIFO is empty, the ASRC_ISRx.TXEMPTY flag rises. If a trigger event occurs while the FIFO is empty, an underrun is generated and the ASRC_ISRx.TXUDR flag rises. The TXUDR flag is cleared on read.

The transmit channel of the ASRC features a DMA channel chunk management. When the number of free spaces reaches the chunk size configured in the field ASRC_CH_CONF.CHUNKx, the ASRC_ISRx.TXCHUNK flag rises, ensuring that chunk size data can be written consecutively. This flag is cleared once the number of data written equals the value configured in ASRC_CH_CONF.CHUNKx.

For each DSP, the chunk size applies to receive and transmit channel management.

The DMA controller transfers must be configured with the same chunk size. Refer to the section “DMA Controller (XDMAC)”.

50.5.5.2 Receive Channel

Each data converted by a DSP is written into the corresponding 32-word depth receive FIFO and can be read in ASRC_RHRx that has been configured through ASRC_CH_CONF.

When at least one data is ready in the receive FIFO, the RXRDY flag rises and remains high as long as there is at least one data to be read in the FIFO.

When a receive FIFO has no more available space, the ASRC_ISRx.RXFULL flag rises. If the DSP writes a new data into the receive FIFO while RXFULL=1, the RXOVR flag rises and the data is not written into the receive FIFO. RXOVR flag is cleared on read.

Once all data contained in the receive FIFO have been read, the ASRC_ISRx.RXEMPTY flag rises. If a data is read while the receive FIFO is empty, an underrun occurs, the RXUDR flag rises. The RXUDR flag is cleared on read.

The receive FIFO features a DMA channel chunk management. When the number of written data reaches the chunk size configured in ASRC_CH_CONF.CHUNK, the ASRC_ISRx.RXCHUNK flag rises, ensuring that chunk size data can be read consecutively. This flag is cleared once the number of data read equals the value configured in ASRC_CH_CONF.CHUNKx.

For each DSP, the chunk size applies to receive and transmit channel management.

The DMA controller transfers must be configured with the same chunk size. Refer to the section “DMA Controller (XDMAC)”.

50.5.5.3 Data Format

The ASRC DSP core datapath is 24 bits wide, thus data size written to the DSP core is always 24 bits. The data written in the ASRC_THRx register can be up to 32 bits.

When the size of the data written to ASRC_THRx register does not match the DSP inherent size (24 bits), the data must be pre-processed. To adapt the potential size difference, the valid number of bits written in ASRC_THRx must be configured in ASRC_VBPS_IN.VBPS_INx. The possible sizes are 8, 10, 12, 14, 16, 18, 20, 24 and 32 bits.

If VBPS_INx is defined as less than or equal to 24 bits, data is first left-shifted by 24 minus VBPS_INx bits. To get full range extension, the extra bits introduced at LSB indexes are the MSB of the valid data (sign bit excluded). The resulting data is written in the DSP core.

If VBPS_INx is defined as 32 bits, data is right-shifted by 8 bits to get the 24 most significant bits before being sent to the DSP core.

Data read from the DSP core is always a 24-bit signed data. Data size read in the ASRC_RHRx register can be defined as 8, 10, 12, 14, 16, 18, 20, 24 and 32 bits in ASRC_VBPS_OUT.VBPS_OUTx. Thus a post-processing must be performed to adapt the size difference.

If VBPS_OUTx is defined as less or equal to 24 bits, data is first right-shifted by 24 minus VBPS_OUTx bits.

If VBPS_OUTx is defined as 32 bits, data is first left-shifted by 8 bits. To get full range extension, the 8 extra bits introduced at LSB indexes are the 8 MSB of the 24-bit data from the DSP (sign bit excluded). The resulting data is written in ASRC_RHRx registers.

Whatever the value configured in VBPS_INx and VBPS_OUTx, each read or write access from/to the system bus carries only one audio data sample.

50.5.6 TDM Audio Stream Management

When a TDM audio stream (more than two audio sources: “2.1”, “5.1” etc.) is provided on a transmit channel, the ASRC automatically distributes the audio sub-streams into several DSPs (maximum 2 sub-streams per DSP). After resampling by DSP, the different sub-streams can be merged into a single receive channel or split into several receive channels.

When several audio input sources of equal or different sampling rates are provided to ASRC through several transmit channels, it is possible to merge the resulting resampled sub-streams into a single TDM audio stream channel.

To manage various cases, ASRC_RHRx and ASRC_THRx can be associated with one or more DSP by configuring ASRC_CH_CONF.THROPMODE and ASRC_CH_CONF.RHROPMODE.

When ASRC_CH_CONF.THROPMODE=0, each ASRC_THRx is associated to only one DSP. In this mode of operation, up to four fully independent stereo streams can be processed. If one or more DSPs are configured in “mono” format, the stream received, processed and written by the DSP is “mono”.

When the ASRC_CH_CONF.THROPMODE=1, data written to ASRC_THR0 are respectively distributed to DSP0 and DSP1. Thus, if both DSP are configured in Stereo mode, two stereo audio data streams are processed and if the ASRC_CH_CONF.RHROPMODE=1, the ASRC_RHR0 provides a resulting, resampled audio stream (TDM frame with four sub-streams). When one ASRC_RHR is configured to read data from multiple DSPs, the output frame rate of the DSP read must be the same. If the ASRC_CH_CONF.RHROPMODE=0, the stereo sub-streams are split between ASRC_RHR0 and ASRC_RHR1 and the output frame rate of each DSP may differ. If DSP1 is configured in mono, the transmit audio stream must be of type “2.1” oriented, that is, three audio sub-streams.

The configuration ASRC_CH_CONF.THROPMODE=1 or ASRC_CH_CONF.RHROPMODE=1 is optimal for processing audio streams embedding three or four audio input sources (for example, “2.1, 2.2, 3.1” type audio streams) and that are transmit via ASRC_THR0 and read via ASRC_RHR0.

When ASRC_CH_CONF.THROPMODE=1 or ASRC_CH_CONF.RHROPMODE=1, ASRC_THR1/RHR1 is dedicated to DSP2 and ASRC_THR2/RHR2 is dedicated to DSP3. Thus, up to two fully independent stereo (or mono) audio streams can be processed at the same time as the audio stream processed by DSP0 and DSP1.

When two audio streams of type “2.1, 2.2, 3.1” must be resampled, ASRC_CH_CONF.THROPMODE/RHROPMODE must be written to 2. ASRC_THR0/RHR0 manages the transfer channels for the first audio stream processed by DSP0 and DSP1 while ASRC_THR1/RHR1 manages the transfer channels for the second audio stream processed by DSP2 and DSP3.

When audio stream embedding more than four and fewer than seven sub-streams (“4.1, 5.1, etc) must be resampled, the ASRC_CH_CONF.THROPMODE/RHROPMODE must be written to 3. ASRC_THR0/RHR0 manages the transfer channels for the audio streams by distributing the audio sub-streams to DSP0, DSP1 and DSP2. DSP3 can be written and read by ASRC_THR1/RHR1 to manage an additional fully independent stereo stream.

For audio stream with seven or eight audio input sources (“6.1”, “7.1”, etc.), the ASRC_CH_CONF.THROPMODE/RHROPMODE must be written to 4. ASRC_THR0/RHR0 manages the transfer channels for the audio streams. Other transmit and receive holding registers have no effect and are not used.

The configuration to associate transmit holding registers to DSPs (ASRC_CH_CONF.THROPMODE) can differ from the configuration to associate receive holding registers to DSPs (ASRC_CH_CONF.RHROPMODE). Depending on the different configurations, the ASRC splits, merges or resamples audio streams.

For all possible configurations, see the figures below.

Figure 50-4. Input and Output Channel Configuration for Operating Modes 0, 1, 2

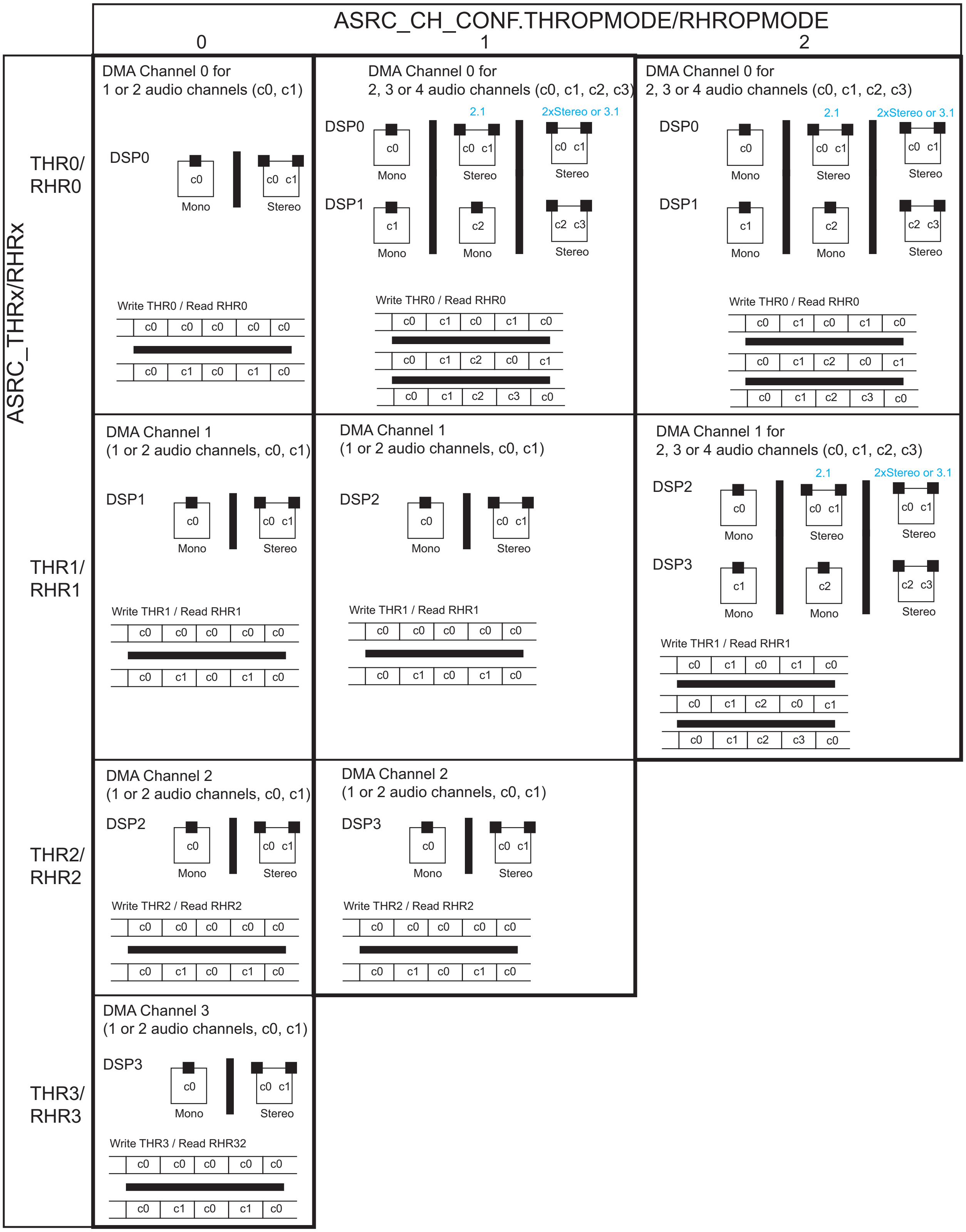
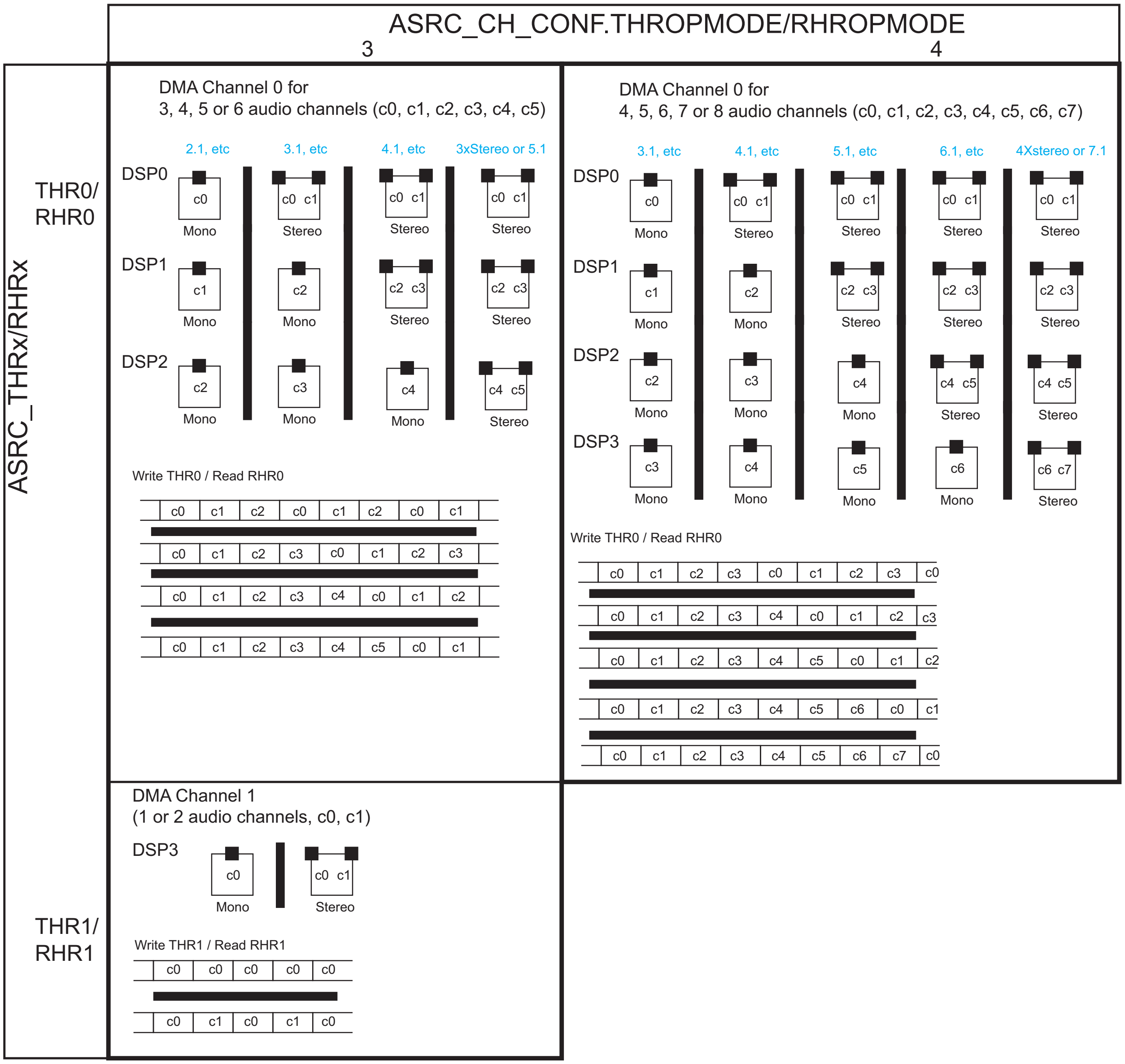


Figure 50-5. Input and Output Channel Configuration for Operating Modes 3, 4



50.5.7 Configuration Limitations

Some limitations exist for the DSP configuration when associating several DSPs to a single transmit or receive holding register.

When ASRC_CH_CONF.RHROPMODE>0, the DSPs sharing the same ASRC_RHRx must have the same output sampling frequency. configured by ASRC_RATIOx.OUTRATIO and ASRC_TRIG.

When ASRC_CH_CONF.THROPMODE>0, the DSPs sharing the same ASRC_THRx must have the same input sampling frequency, configured by ASRC_RATIOx.INRATIO and ASRC_TRIG.

When ASRC_CH_CONF.RHROPMODE>0 or ASRC_CH_CONF.THROPMODE>0, the mono or stereo configuration of DSPs are limited to the values described in the figures shown above.

The configuration of DMA transfer chunk size has some restrictions listed in the table below. The DMA receives fewer requests to transfer data when the chunk size is high. Therefore, it reduces the overhead processing for each transfer request and leads to improved performance at system level.

Major configuration errors are reported in ASRC_ESR. For details, see ASRC_ESR.

Table 50-2. DMA Chunk Size Configuration Limitations

	ASRC_CH_CONF.THROPMODE / ASRC_CH_CONF.RHROPMODE					
Value	0		1,2,3[(1)]		4	
DSP	Mono	Stereo	Mono	Stereo	Mono	Stereo
CHUNKx	1	1	1	1	1	1
	2	2	2	2	2	2
	4	4	4	4	4	4
	Forbidden	8	8	8	8	8
	Forbidden	Forbidden	Forbidden	16	16	16

Note:
1. Applies only to ASRC_RHRx/THRx sharing a DSP. The chunk limitation for ASRC_RHRx/THRx driving one DSP is the same as the limitation for THROPMODE/RHROPMODE=0.

50.5.8 Initialization Sequence

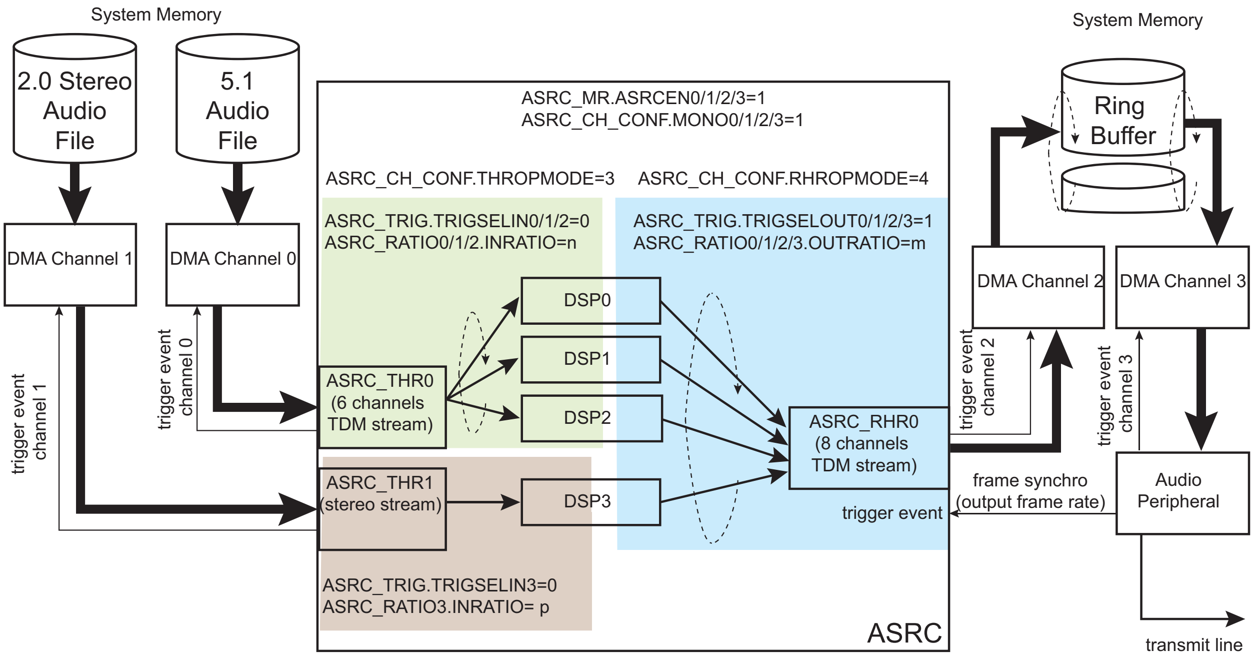
In order to use one DSP, the user must follow the sequence below:

1. Select the input and output triggers. Trigger events (from audio peripherals) must be stable before the configuration/selection in the ASRC.
2. Define the DSP configuration, Channel Transmitting and Receiving modes.
3. Enable the DSP or group of DSPs that will be used (ASRC_MR.ASRCENx=1).
4. Wait for the lock flag of DSPs. If several DSPs share the same holding register, wait for all DSPs lock status.
5. Start the data transfer either by software, or enable the DMA transfers to transmit and receive holding register.
6. ASRC_ESR must read 0.

50.5.9 Application Example

The figure below illustrates a use case where two audio files are stored in system memory, each file sampled at different sample rates, transferred to the ASRC via two DMA channels and resampled at another sampling rate before being transmitted on the output audio line via an audio peripheral linked to the ASRC via a trigger event to provide the output frame rate. Received audio streams are merged into a single output stream.

Figure 50-6. Application Example



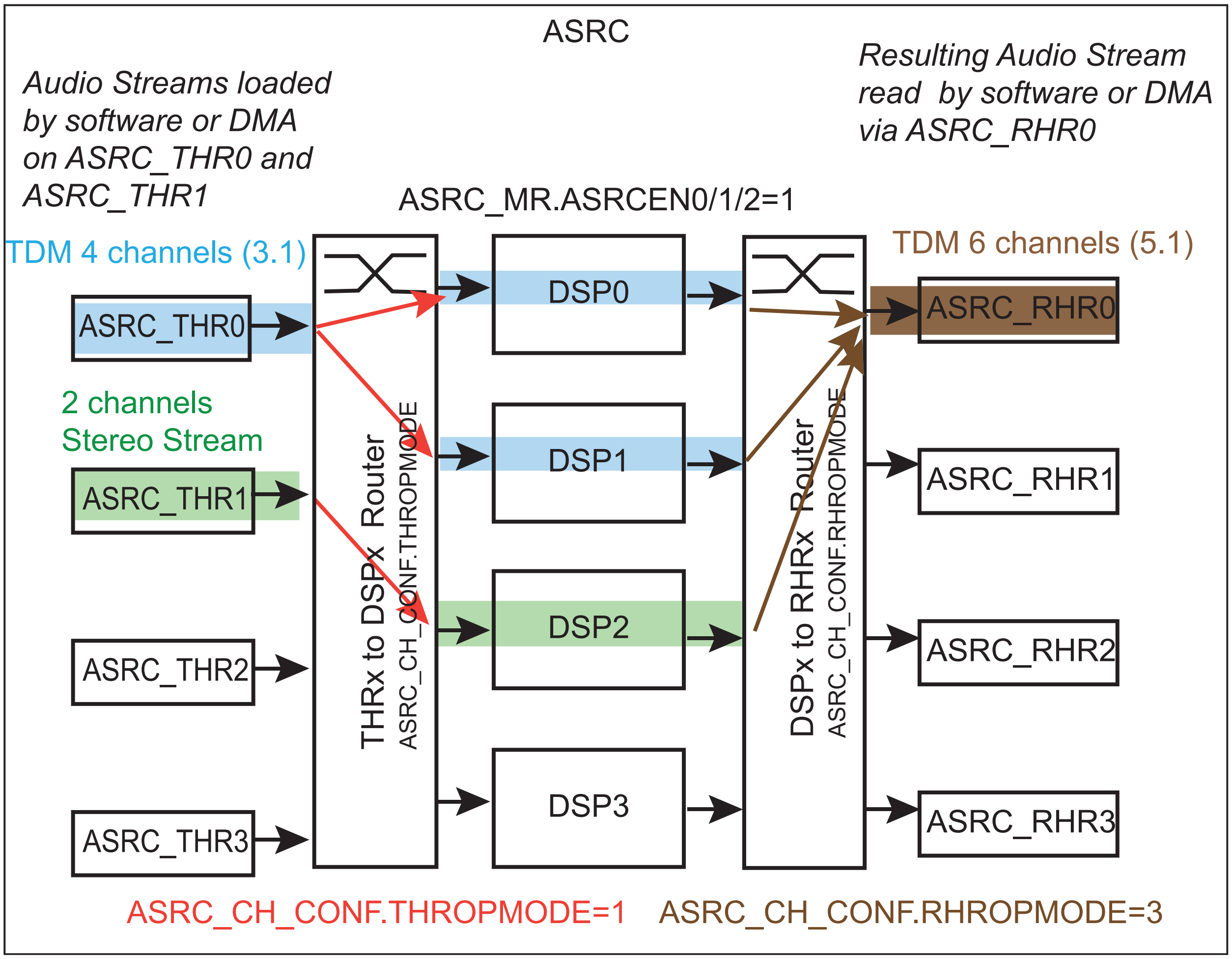
50.5.10 Channels Split/Merge Capability

The ASRC provides a capability to merge several audio streams into one or more resulting streams and to split an incoming multichannel audio stream into several streams.

The merge or split operation results from the configuration of ASRC_CH_CONF.THROPMODE and ASRC_CH_CONF.RHROPMODE.

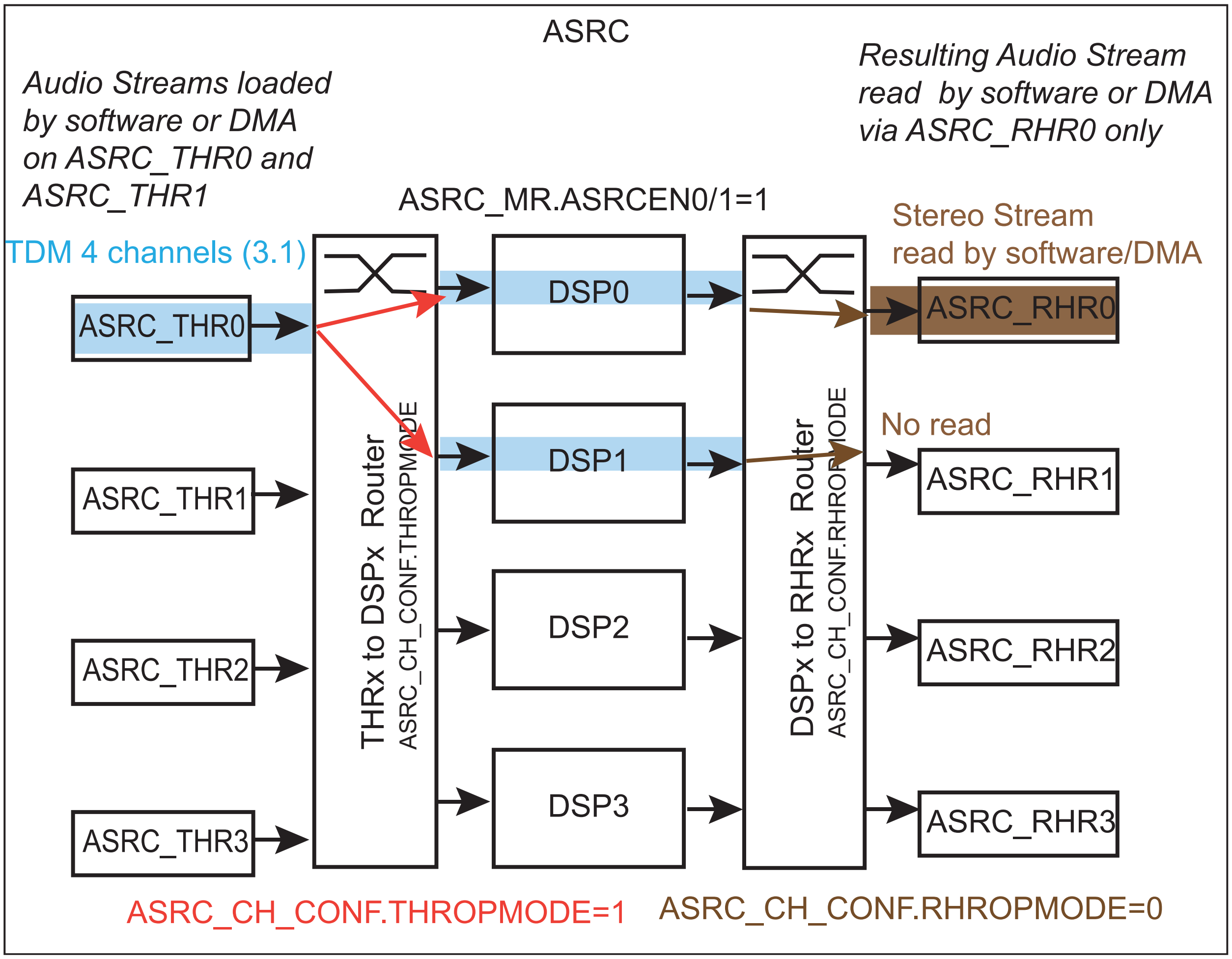
The merge operation adds one or more channels to an incoming audio stream when the resulting converted stream is read via a single ASRC_RHR register (see the following figure).

Figure 50-7. Audio Streams Merge Operation Example



The split operation drops one or more channels from an incoming audio stream when the resulting converted stream is routed via several ASRC_RHRx registers. When some ASRC_RHRx are not read, a drop operation occurs on the audio streams routed on these registers (see the following figure).

Figure 50-8. Audio Stream Split Operation Example



50.5.11 Register Write Protection

To prevent any single software error from corrupting ASRC behavior, certain registers in the address space can be write-protected by setting the WPEN or the WPITEN bit in the Write Protection Mode Register (ASRC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status Register (ASRC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading ASRC_WPSR.

The following registers are write-protected when the WPEN bit is set in ASRC_WPMR:

- ASRC Mode Register
- ASRC Ratio Register of Stereo Channel x
- ASRC Valid bit Per Sample In Register
- ASRC Valid bit Per Sample Out Register
- ASRC Channel Configuration Register
- ASRC Trigger Selection Register

The following registers are write-protected when the WPITEN bit is set in ASRC_WPMR:

- ASRC Interrupt Enable Register of Stereo Channel x
- ASRC Interrupt Disable Register of Stereo Channel x

The following register is write-protected when the WPCREN bit is set in ASRC_WPMR:

- ASRC Control Register

50.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ASRC_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	ASRC_MR	31:24								
		23:16								
		15:8				GT96K				
		7:0					ASRCEN3	ASRCEN2	ASRCEN1	ASRCEN0
0x08	ASRC_RATIO0	31:24	OUTRATIO[15:8]							
		23:16	OUTRATIO[7:0]							
		15:8	INRATIO[15:8]							
		7:0	INRATIO[7:0]							
0x0C	ASRC_RATIO1	31:24	OUTRATIO[15:8]							
		23:16	OUTRATIO[7:0]							
		15:8	INRATIO[15:8]							
		7:0	INRATIO[7:0]							
0x10	ASRC_RATIO2	31:24	OUTRATIO[15:8]							
		23:16	OUTRATIO[7:0]							
		15:8	INRATIO[15:8]							
		7:0	INRATIO[7:0]							
0x14	ASRC_RATIO3	31:24	OUTRATIO[15:8]							
		23:16	OUTRATIO[7:0]							
		15:8	INRATIO[15:8]							
		7:0	INRATIO[7:0]							
0x18	ASRC_VBPS_IN	31:24					VBPS_IN3[3:0]			
		23:16					VBPS_IN2[3:0]			
		15:8					VBPS_IN1[3:0]			
		7:0					VBPS_IN0[3:0]			
0x1C	ASRC_VBPS_OUT	31:24					VBPS_OUT3[3:0]			
		23:16					VBPS_OUT2[3:0]			
		15:8					VBPS_OUT1[3:0]			
		7:0					VBPS_OUT0[3:0]			
0x20	ASRC_CH_CONF	31:24		CHUNK3[2:0]				CHUNK2[2:0]		
		23:16		CHUNK1[2:0]				CHUNK0[2:0]		
		15:8					MONO3	MONO2	MONO1	MONO0
		7:0		RHROPMODE[2:0]				THROPMODE[2:0]		
0x24	ASRC_TRIG	31:24	TRIGSELOUT3[3:0]				TRIGSELOUT2[3:0]			
		23:16	TRIGSELOUT1[3:0]				TRIGSELOUT0[3:0]			
		15:8	TRIGSELIN3[3:0]				TRIGSELIN2[3:0]			
		7:0	TRIGSELIN1[3:0]				TRIGSELIN0[3:0]			
0x28	ASRC_RHR0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C	ASRC_RHR1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x30	ASRC_RHR2	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x34	ASRC_RHR3	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38 ... 0x47	Reserved									
0x48	ASRC_THR0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x4C	ASRC_THR1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x50	ASRC_THR2	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x54	ASRC_THR3	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x58 ... 0x67	Reserved									
0x68	ASRC_IER0	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x6C	ASRC_IER1	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x70	ASRC_IER2	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x74	ASRC_IER3	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x78	ASRC_IDR0	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x7C	ASRC_IDR1	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x80	ASRC_IDR2	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x84	ASRC_IDR3	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x88	ASRC_IMR0	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x8C	ASRC_IMR1	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x90	ASRC_IMR2	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x94	ASRC_IMR3	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x98	ASRC_ISR0	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0x9C	ASRC_ISR1	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0xA0	ASRC_ISR2	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0xA4	ASRC_ISR3	31:24		LOCK						
		23:16								SECE
		15:8			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
		7:0			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
0xA8	ASRC_ESR	31:24	DAOERRx3	DAOERRx2	DAOERRx1	DAOERRx0	DAIERRx3	DAIERRx2	DAIERRx1	DAIERRx0
		23:16	OTRIGERR	ITRIGERR	DARHXERR	DATHXERR	DAPSELORE DERR	DAPSELIRED ERR	DAPSELOMA XERR	DAPSELIMAX ERR
		15:8	DERR			OUTCFGERR[4:0]				
		7:0				INCFGERR[4:0]				
0xAC ... 0xE3	Reserved									
0xE4	ASRC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	ASRC_WPSR	31:24							SWETYP[1:0]	
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0					SWE	SEQE		WPVS

50.6.1 ASRC Control Register

Name: ASRC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the ASRC.

50.6.2 ASRC Mode Register

Name: ASRC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				GT96K				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
					ASRCEN3	ASRCEN2	ASRCEN1	ASRCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 12 – GT96K Frequency Sampling Greater Than 96 kHz

Value	Name	Description
0	DISABLE	Up to 4 DSPs with up to 96 kHz as the upper bound limit of the sampling frequency.
1	ENABLE	Up to 2 DSPs with up to 192 kHz as the upper bound limit of the sampling frequency. The number of enabled DSPs is limited to 2.

Bits 0, 1, 2, 3 – ASRCENx ASRC Stereo Channel x Enable

Value	Name	Description
0	DISABLE	DSPx is disabled.
1	ENABLE	DSPx is enabled.

50.6.3 ASRC Ratio Register of Stereo Channel x

Name: ASRC_RATIOx
Offset: 0x08 + x*0x04 [x=0..3]
Reset: 0x04000400
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
	OUTRATIO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit	23	22	21	20	19	18	17	16
	OUTRATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INRATIO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
	INRATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – OUTRATIO[15:0] Output Internal Sampling Rate Ratio
The output internal sampling rate is defined by a division ratio applied on the GCLK:
$F_s(out) = F_{GCLK}/OUTRATIO$
The minimum ratio value is 1024 or 2048 (see Limitations).

Bits 15:0 – INRATIO[15:0] Input Internal Sampling Rate Ratio
The input internal sampling rate is defined by a division ratio applied on the GCLK:
$F_s(in) = F_{GCLK}/INRATIO$
The minimum ratio value is 1024 or 2048 (see Limitations).

50.6.4 ASRC Valid bit Per Sample In Register

Name: ASRC_VBPS_IN
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
					VBPS_IN3[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VBPS_IN2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					VBPS_IN1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					VBPS_IN0[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0:3, 8:11, 16:19, 24:27 – VBPS_INx Valid Bit Per Sample In of DSP x
Defines the number of valid bits of the incoming data of DSP x.

Value	Name	Description
0	8_BIT	The 8 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [15:7] of the 24-bit data are driven by the 8 LSB of THR registers to increase the dynamic range of the data sent to DSP. The bits [7:0] are driven by 0s.
1	16_BIT	The 16 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [7:0] of the 24-bit data are driven by the bits [15:8] of the THR registers to increase the dynamic range of the data sent to DSP.
2	20_BIT	The 20 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [3:0] of the 24-bit data are driven by the bits [19:16] of the THR registers to increase the dynamic range of the data sent to DSP.
3	24_BIT	The 24 LSB of THR registers are entered to the DSP.
4	32_BIT	The 24 MSB of THR registers are entered to the DSP.
5	10_BIT	The 10 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [13:4] of the 24-bit data are driven by the 10 LSB of THR registers to increase the dynamic range of the data sent to DSP. The bits [3:0] are driven by 0s.
6	12_BIT	The 12 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [11:0] of the 24-bit data are driven by the bits [11:0] of the THR registers to increase the dynamic range of the data sent to DSP.
7	14_BIT	The 14 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [9:0] of the 24-bit data are driven by the bits [13:4] of the THR registers to increase the dynamic range of the data sent to DSP.

Value	Name	Description
8	18_BIT	The 18 LSB of THR registers are left aligned on 24-bit data entered to the DSP. The bits [5:0] of the 24-bit data are driven by the bits [17:12] of the THR registers to increase the dynamic range of the data sent to DSP.

50.6.5 ASRC Valid bit Per Sample Out Register

Name: ASRC_VBPS_OUT
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
					VBPS_OUT3[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VBPS_OUT2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					VBPS_OUT1[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					VBPS_OUT0[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0:3, 8:11, 16:19, 24:27 – VBPS_OUTx Valid Bit Per Sample Out of DSP x
Defines the number of valid bits of the outgoing data of DSP x.

Value	Name	Description
0	8_BIT	The 8 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
1	16_BIT	The 16 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
2	20_BIT	The 20 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
3	24_BIT	The 24 LSB of RHR registers are read from the 24-bit DSP output.
4	32_BIT	The 24 MSB of RHR registers are the 24-bit output of the DSP. The 8 LSB of the RHR are driven by bits [23:17] of the DSP output.
5	10_BIT	The 10 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
6	12_BIT	The 12 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
7	14_BIT	The 14 MSB of the 24-bit read from the DSP output are right-aligned on RHR.
8	18_BIT	The 18 MSB of the 24-bit read from the DSP output are right-aligned on RHR.

50.6.6 ASRC Channel Configuration Register

Name: ASRC_CH_CONF
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
		CHUNK3[2:0]				CHUNK2[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		CHUNK1[2:0]				CHUNK0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
					MONO3	MONO2	MONO1	MONO0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RHROPMODE[2:0]				THROPMODE[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 16:18, 20:22, 24:26, 28:30 – CHUNKx DMA DSP x CHUNK Size
DMA channel chunk size for Receive and Transmit FIFOs of DSP x.
When Receive FIFO of DSP x has CHUNK unread data, the flag ASRC_ISRx.RXCHUNK rises.
When Transmit FIFO of DSP x has CHUNK empty data, the flag ASRC_ISRx.TXCHUNK rises.

Value	Name	Description
0	1_DATA	The DMA chunk size must be configured to transfer 1 data for each rising edge of the trigger event.
1	2_DATA	The DMA chunk size must be configured to transfer 2 data for each rising edge of the trigger event.
2	4_DATA	The DMA chunk size must be configured to transfer 4 data for each rising edge of the trigger event.
3	8_DATA	The DMA chunk size must be configured to transfer 8 data for each rising edge of the trigger event.
4	16_DATA	The DMA chunk size must be configured to transfer 16 data for each rising edge of the trigger event.

Bits 8, 9, 10, 11 – MONOx DSP x Mono Operating Mode

Value	Name	Description
0	DISABLED	DSP operates in Stereo mode.
1	ENABLED	DSP operates in Mono mode.

Bits 6:4 – RHROPMODE[2:0] Receive Holding Registers Operating Mode
Optimizes the DMA channels versus the number and type of audio streams to manage
For more details, see DSP and Channel Configuration.

Value	Name	Description
0	RHRX_UPTO_2CH	The ASRC_RHRx can receive up to 2 audio streams (e.g. 1 stereo channel for each ASRC_RHR). Each ASRC_RHRx is routed on each Stereo DSP x.
1	RHR0_UPTO_4CH	The ASRC_RHR0 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_RHR0). The ARSC_RHR1 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_RHR1). The ARSC_RHR2 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_RHR2). ASRC_RHR3 must not be used.
2	RHR01_UPTO_4CH	The ASRC_RHR0 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_RHR0). The ASRC_RHR1 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_RHR1). ASRC_RHR2 and ASRC_RHR3 must not be used.
3	RHR0_UPTO_6CH	The ASRC_RHR0 can receive up to 6 audio streams (e.g. up to 3 stereo channels on ASRC_RHR0). The ARSC_RHR1 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_RHR1) ASRC_RHR2 and ASRC_RHR3 must not be used.
4	RHR0_UPTO_8CH	The ASRC_RHR0 can receive up to 8 audio streams (e.g. up to 4 stereo channels on ASRC_RHR0). ASRC_RHR1, ASRC_RHR2 and ASRC_RHR3 must not be used.

Bits 2:0 – THROPMODE[2:0] Transmit Holding Registers Operating Mode
Optimizes the DMA channels versus the number and type of audio streams to manage.
For more details, see DSP and Channel Configuration.

Value	Name	Description
0	THRX_UPTO_2CH	The ASRC_THRx can receive up to 2 audio streams (e.g. 1 stereo channel for each ASRC_THR). Each ASRC_THRx is routed on each Stereo DSP x.
1	THR0_UPTO_4CH	The ASRC_THR0 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_THR0). The ARSC_THR1 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_THR1). The ARSC_THR2 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_THR2). ASRC_THR3 must not be used.
2	THR01_UPTO_4CH	The ASRC_THR0 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_THR0). The ASRC_THR1 can receive up to 4 audio streams (e.g. up to 2 stereo channels on ASRC_THR1). ASRC_THR2 and ASRC_THR3 must not be used.
3	THR0_UPTO_6CH	The ASRC_THR0 can receive up to 6 audio streams (e.g. up to 3 stereo channels on ASRC_THR0). The ARSC_THR1 can receive up to 2 audio streams (e.g. 1 stereo channel on ASRC_THR1) ASRC_THR2 and ASRC_THR3 must not be used.

Value	Name	Description
4	THR0_UPTO_8CH	The ASRC_THR0 can receive up to 8 audio streams (e.g. up to 4 stereo channels on ASRC_THR0). ASRC_THR1,ASRC_THR2 and ASRC_RHR3 must not be used.

50.6.7 ASRC Trigger Selection Register

Name: ASRC_TRIG
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ASRC_WPMR.

Bit	31	30	29	28	27	26	25	24
	TRIGSELOUT3[3:0]				TRIGSELOUT2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TRIGSELOUT1[3:0]				TRIGSELOUT0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TRIGSELIN3[3:0]				TRIGSELIN2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TRIGSELIN1[3:0]				TRIGSELIN0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16:19, 20:23, 24:27, 28:31 – TRIGSELOUTx Output Trigger Source Selection of Channel x

Value	Name	Description
0	GCLKDIV	Internal trigger event, a configurable division of GCLK (see ASRC_RATIOx for divider).
1	I2SMCC0 ws	Trigger event from audio peripheral.
2	I2SMCC1 ws	Trigger event from audio peripheral.
3	PDMC0 data synchro	Trigger event from audio peripheral.
4	PDMC1 data synchro	Trigger event from audio peripheral.
5	SSC0 RF	Trigger event from audio peripheral.
6	SSC0 TF	Trigger event from audio peripheral.
7	SSC1 RF	Trigger event from audio peripheral.

Bits 0:3, 4:7, 8:11, 12:15 – TRIGSELINx Input Trigger Source Selection of Channel x

Value	Name	Description
0	GCLKDIV	Internal trigger event, a configurable division of GCLK (see ASRC_RATIOx for divider).
1	I2SMCC0 ws	Trigger event from audio peripheral.
2	I2SMCC1 ws	Trigger event from audio peripheral.
3	PDMC0 data synchro	Trigger event from audio peripheral.
4	PDMC1 data synchro	Trigger event from audio peripheral.
5	SSC0 RF	Trigger event from audio peripheral.
6	SSC0 TF	Trigger event from audio peripheral.
7	SSC1 RF	Trigger event from audio peripheral.

50.6.8 ASRC Receive Holding Register of Stereo Channel x

Name: ASRC_RHRx
Offset: 0x28 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Converted Data
Contains the converted data.

50.6.9 ASRC Transmit Holding Register of Channel x

Name: ASRC_THRx
Offset: 0x48 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – DATA[31:0] Data to Convert
Contains the data to be converted.

50.6.10 ASRC Interrupt Enable Register of Stereo Channel x

Name: ASRC_IERx
Offset: 0x68 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in ASRC_WPMR.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
		LOCK						
Access		W						
Reset		–						

Bit	23	22	21	20	19	18	17	16
								SECE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 30 – LOCK DPLL Locked Interrupt Enable

Bit 16 – SECE Security/Safety Report Enable

Bit 13 – TXOVR Transmit Over Flow Interrupt Enable

Bit 12 – TXUDR Transmit Under Flow Interrupt Enable

Bit 11 – TXCHUNK Transmit FIFO Chunk Interrupt Enable

Bit 10 – TXFULL Transmit FIFO Full Interrupt Enable

Bit 9 – TXEMPTY Transmit FIFO Empty Interrupt Enable

Bit 8 – TXRDY Transmit Ready Interrupt Enable

Bit 5 – RXOVR Receive Over Flow Interrupt Enable

Bit 4 – RXUDR Receive Under Flow Interrupt Enable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Enable

Bit 2 – RXFULL Receive FIFO Full Interrupt Enable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Enable

Bit 0 – RXRDY Receive Ready Interrupt Enable

50.6.11 ASRC Interrupt Disable Register of Stereo Channel x

Name: ASRC_IDRx
Offset: 0x78 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in ASRC_WPMR.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
		LOCK						
Access		W						
Reset		–						

Bit	23	22	21	20	19	18	17	16
								SECE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 30 – LOCK DPLL Locked Interrupt Disable

Bit 16 – SECE Security/Safety Report Disable

Bit 13 – TXOVR Transmit Over Flow Interrupt Disable

Bit 12 – TXUDR Transmit Under Flow Interrupt Disable

Bit 11 – TXCHUNK Transmit FIFO Chunk Interrupt Disable

Bit 10 – TXFULL Transmit FIFO Full Interrupt Disable

Bit 9 – TXEMPTY Transmit FIFO Empty Interrupt Disable

Bit 8 – TXRDY Transmit Ready Interrupt Disable

Bit 5 – RXOVR Receive Over Flow Interrupt Disable

Bit 4 – RXUDR Receive Under Flow Interrupt Disable

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Disable

Bit 2 – RXFULL Receive FIFO Full Interrupt Disable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Disable

Bit 0 – RXRDY Receive Ready Interrupt Disable

50.6.12 ASRC Interrupt Mask Register of Stereo Channel x

Name: ASRC_IMRx
Offset: 0x88 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
		LOCK						
Access		R						
Reset		0						

Bit	23	22	21	20	19	18	17	16
								SECE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 30 – LOCK DPLL Locked Interrupt Mask

Bit 16 – SECE Security/Safety Report Mask

Bit 13 – TXOVR Transmit Over Flow Interrupt Mask

Bit 12 – TXUDR Transmit Under Flow Interrupt Mask

Bit 11 – TXCHUNK Transmit FIFO Chunk Interrupt Mask

Bit 10 – TXFULL Transmit FIFO Full Interrupt Mask

Bit 9 – TXEMPTY Transmit FIFO Empty Interrupt Mask

Bit 8 – TXRDY Transmit Ready Interrupt Mask

Bit 5 – RXOVR Receive Over Flow Interrupt Mask

Bit 4 – RXUDR Receive Under Flow Interrupt Mask

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Mask

Bit 2 – RXFULL Receive FIFO Full Interrupt Mask

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Mask

Bit 0 – RXRDY Receive Ready Interrupt Mask

50.6.13 ASRC Interrupt Status Register of Stereo Channel x

Name: ASRC_ISRx
Offset: 0x98 + x*0x04 [x=0..3]
Reset: 0x80000302
Property: Read-only

The following values are valid for all listed bit names of this register:

0: The corresponding interrupt source is not active.

1: The corresponding interrupt source is active.

Bit	31	30	29	28	27	26	25	24
		LOCK						
Access		R						
Reset		0						

Bit	23	22	21	20	19	18	17	16
								SECE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
			TXOVR	TXUDR	TXCHUNK	TXFULL	TXEMPTY	TXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	1	1

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR	RXCHUNK	RXFULL	RXEMPTY	RXRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	1	0

Bit 30 – LOCK DPLL Locked Interrupt Status (cleared by writing ASRC_MR.ASRCENx=0

Bit 16 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event has occurred since the last read of ASRC_ISR.
1	One or more safety or security events have occurred since the last read of ASRC_ISR. For details on the event, refer to ASRC_WPSR.

Bit 13 – TXOVR Transmit Over Flow Interrupt Status (cleared on read)

Bit 12 – TXUDR Transmit Under Flow Interrupt Status (cleared on read)

Bit 11 – TXCHUNK Transmit FIFO Chunk Interrupt Status

Bit 10 – TXFULL Transmit FIFO Full Interrupt Status

Bit 9 – TXEMPTY Transmit FIFO Empty Interrupt Status

Bit 8 – TXRDY Transmit Ready Interrupt Status

Bit 5 – RXOVR Receive Over Flow Interrupt Status (cleared on read)

Bit 4 – RXUDR Receive Under Flow Interrupt Status (cleared on read)

Bit 3 – RXCHUNK Receive FIFO Chunk Interrupt Status

Bit 2 – RXFULL Receive FIFO Full Interrupt Status

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Status

Bit 0 – RXRDY Receive Ready Interrupt Status

50.6.14 ASRC Error Status Register

Name: ASRC_ESR
Offset: 0xA8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DAOERRx3	DAOERRx2	DAOERRx1	DAOERRx0	DAIERRx3	DAIERRx2	DAIERRx1	DAIERRx0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OTRIGERR	ITRIGERR	DARHXERR	DATHXERR	DAPSELOREDERR	DAPSELIREDERR	DAPSELOMAXERR	DAPSELIMAXERR
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DERR			OUTCFGERR[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				INCFGERR[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 28, 29, 30, 31 – DAOERRx Direct Access Output Transfer Error Status (cleared on read)

Bits 24, 25, 26, 27 – DAIERRx Direct Access Input Transfer Error Status (cleared on read)

Bit 23 – OTRIGERR Output Trigger Configuration Error

Value	Description
0	Validates the configuration of ASRC_TRIG.TRIGSELINx versus ASRC_CH_CONF.RHROPMODE.
1	The configuration of ASRC_TRIG.TRIGSELINx is incompatible with the configuration of ASRC_CH_CONF.RHROPMODE. The error is only reported when 1 DSP is enabled (ASRC_MR.ASRCENx differs from 0). The error is cleared when the configuration is correct.

Bit 22 – ITRIGERR Input Trigger Configuration Error

Value	Description
0	Validates the configuration of ASRC_TRIG.TRIGSELINx versus ASRC_CH_CONF.THROPMODE.
1	The configuration of ASRC_TRIG.TRIGSELINx is incompatible with the configuration of ASRC_CH_CONF.THROPMODE. The error is only reported when 1 DSP is enabled (ASRC_MR.ASRCENx differs from 0). The error is cleared when the configuration is correct.

Bit 21 – DARHXERR Direct Access DARHM/DARHRx Configuration Error

Value	Description
0	Validates the configuration of ASRC_MR.DARHM/DRHRx.
1	The configuration of ASRC_MR.DARHM/DRHRx is incompatible with the configuration of ASRC_CH_CONF.RHROPMODE. The error is only reported when 1 DSP is enabled (ASRC_MR.ASRCENx differs from 0). The error is cleared when the configuration is correct.

Bit 20 – DATHXERR Direct Access DATHM/DATHRx Configuration Error

Value	Description
0	Validates the configuration of ASRC_MR.DATHM/DTHRx.

Value	Description
1	The configuration of ASRC_MR.DATHM/DTHRx is incompatible with the configuration of ASRC_CH_CONF.THROPMODE. The error is only reported when 1 DSP is enabled (ASRC_MR.ASRCENx differs from 0). The error is cleared when the configuration is correct.

Bit 19 – DAPSELOREDERR Direct Access Output Peripheral Selection Redundancy Configuration Error

Value	Description
0	Validates the configuration of ASRC_DAPSELR.OUT_CHx and ASRC_MR.DARHRx.
1	The same audio peripheral is enabled to transmit data into more than one direct access channel. The ASRC_DAPSELR.OUT_CHx and ASRC_MR.DARHRx must be correctly configured to clear the error.

Bit 18 – DAPSELIREDERR Direct Access Input Peripheral Selection Redundancy Configuration Error

Value	Description
0	Validates the configuration of ASRC_DAPSELR.IN_CHx and ASRC_MR.DATHRx.
1	The same audio peripheral is enabled to transmit data into more than one direct access channel. The ASRC_DAPSELR.IN_CHx and ASRC_MR.DATHRx must be correctly configured to clear the error.

Bit 17 – DAPSELOMAXERR Direct Access Output Peripheral Selection Configuration Error on Maximum Values

Value	Description
0	None of the values in ASRC_DAPSELR.OUT_CHx fields overpass the maximum number of audio peripherals allowed to transfer data directly to the ASRC or the related direct access transmit channel is disabled in ASRC_MR.DARHRx.
1	One of the values in ASRC_DAPSELR.OUT_CHx fields overpass the maximum number of audio peripherals allowed to transfer data directly to the ASRC and the related direct access transmit channel is enabled in ASRC_MR.DARHRx. The error is cleared when the configuration is correct.

Bit 16 – DAPSELIMAXERR Direct Access Input Peripheral Selection Configuration Error on Maximum Values

Value	Description
0	None of the values in ASRC_DAPSELR.IN_CHx fields overpass the maximum number of audio peripherals allowed to transfer data directly to the ASRC or the related direct access transmit channel is disabled in ASRC_MR.DATHRx.
1	One of the values in ASRC_DAPSELR.IN_CHx fields overpass the maximum number of audio peripherals allowed to transfer data directly to the ASRC and the related direct access transmit channel is enabled in ASRC_MR.DATHRx. The error is cleared when the configuration is correct.

Bit 15 – DERR DSP Overflow Error

Value	Description
0	No DSP overflow error detected.
1	The sampling frequency overpasses the value allowed by the configuration.

Bits 12:8 – OUTCFGERR[4:0] Output Configuration Error

Value	Name	Description
0	OUTCFG_OK	Correct configuration of ASRC_MR, ASRC_VBPS_OUT, ASRC_CH_CONF.
1	OP0_D0_CHK16_8M	Channel operating mode 0, DSP0: CHUNK0=16 or CHUNK0=8 and MONO0=1 is not supported.
2	OP0_D1_CHK16_8M	Channel operating mode 0, DSP1: CHUNK1=16 or CHUNK1=8 and MONO1=1 is not supported.
3	OP0_D2_CHK16_8M	Channel operating mode 0, DSP2: CHUNK2=16 or CHUNK2=8 and MONO2=1 is not supported.
4	OP0_D3_CHK16_8M	Channel operating mode 0, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3=1 is not supported.
5	OP1_D01_EN	Channel operating mode 1, DSP0/1: enable configuration is not correct (1 DSP is disabled)
6	OP1_D01_M	Channel operating mode 1, DSP0/1: MONO0=1, MONO1=0 is not supported.
7	OP1_D01_CHK16	Channel operating mode 1, DSP0/1: CHUNK0=16 is not supported.
8	OP1_D2_CHK16_8M	Channel operating mode 1, DSP2: CHUNK2=16 or CHUNK2=8 and MONO2=1 is not supported.

Value	Name	Description
9	OP1_D3_CHK16_8M	Channel operating mode 1, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3=1 is not supported.
10	OP2_D01_EN	Channel operating mode 2, DSP0/1: enable configuration is not correct (1 DSP is disabled)
11	OP2_D01_M	Channel operating mode 2, DSP0/1: MONO0=1, MONO1=0 is not supported.
12	OP2_D01_CHK16	Channel operating mode 2, DSP0/1: CHUNK0=16 is not supported.
13	OP2_D23_EN	Channel operating mode 2, DSP2/3: enable configuration is not correct (1 DSP is disabled)
14	OP2_D23_M	Channel operating mode 2, DSP2/3: MONO2=1, MONO3=0 is not supported.
15	OP2_D23_CHK16	Channel operating mode 2, DSP2/3: CHUNK2=16 is not supported.
16	OP3_D012_EN	Channel operating mode 3, DSP0/1/2: enable configuration is not correct (at least 1 DSP is disabled)
17	OP3_D012_M	Channel operating mode 3, DSP0/1/2: mono configuration is not correct (e.g. MONO0=1, MONO1=0 or MONO2=0 is not supported).
18	OP3_D012_CHK16	Channel operating mode 3, DSP0/1/2: CHUNK0=16 is not supported.
19	OP3_D3_CHK16_8M	Channel operating mode 3, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3 is not supported.
20	OP4_D0123_EN	Channel operating mode 4, DSP0/1/2/3: enable configuration is not correct (at least 1 DSP is disabled)
21	OP4_D0123_M	Channel operating mode 4, DSP0/1/2/3: mono configuration is not correct (e.g. MONO0=1, MONO1=0 or MONO2=0 or MONO3=0 is not supported).

Bits 4:0 – INCFGERR[4:0] Input Configuration Error

Value	Name	Description
0	OUTCFG_OK	Correct configuration of ASRC_MR, ASRC_VBPS_OUT, ASRC_CH_CONF.
1	OP0_D0_CHK16_8M	Channel operating mode 0, DSP0: CHUNK0=16 or CHUNK0=8 and MONO0=1 is not supported.
2	OP0_D1_CHK16_8M	Channel operating mode 0, DSP1: CHUNK1=16 or CHUNK1=8 and MONO1=1 is not supported.
3	OP0_D2_CHK16_8M	Channel operating mode 0, DSP2: CHUNK2=16 or CHUNK2=8 and MONO2=1 is not supported.
4	OP0_D3_CHK16_8M	Channel operating mode 0, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3=1 is not supported.
5	OP1_D01_EN	Channel operating mode 1, DSP0/1: enable configuration is not correct (1 DSP is disabled)
6	OP1_D01_M	Channel operating mode 1, DSP0/1: MONO0=1, MONO1=0 is not supported.
7	OP1_D01_CHK16	Channel operating mode 1, DSP0/1: CHUNK0=16 is not supported.
8	OP1_D2_CHK16_8M	Channel operating mode 1, DSP2: CHUNK2=16 or CHUNK2=8 and MONO2=1 is not supported.
9	OP1_D3_CHK16_8M	Channel operating mode 1, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3=1 is not supported.
10	OP2_D01_EN	Channel operating mode 2, DSP0/1: enable configuration is not correct (1 DSP is disabled)
11	OP2_D01_M	Channel operating mode 2, DSP0/1: MONO0=1, MONO1=0 is not supported.
12	OP2_D01_CHK16	Channel operating mode 2, DSP0/1: CHUNK0=16 is not supported.
13	OP2_D23_EN	Channel operating mode 2, DSP2/3: enable configuration is not correct (1 DSP is disabled)
14	OP2_D23_M	Channel operating mode 2, DSP2/3: MONO2=1, MONO3=0 is not supported.
15	OP2_D23_CHK16	Channel operating mode 2, DSP2/3: CHUNK2=16 is not supported.
16	OP3_D012_EN	Channel operating mode 3, DSP0/1/2: enable configuration is not correct (at least 1 DSP is disabled)
17	OP3_D012_M	Channel operating mode 3, DSP0/1/2: mono configuration is not correct (e.g. MONO0=1, MONO1=0 or MONO2=0 is not supported).
18	OP3_D012_CHK16	Channel operating mode 3, DSP0/1/2: CHUNK0=16 is not supported.
19	OP3_D3_CHK16_8M	Channel operating mode 3, DSP3: CHUNK3=16 or CHUNK3=8 and MONO3 is not supported.

Value	Name	Description
20	OP4_D0123_EN	Channel operating mode 4, DSP0/1/2/3: enable configuration is not correct (at least 1 DSP is disabled)
21	OP4_D0123_M	Channel operating mode 4, DSP0/1/2/3: mono configuration is not correct (e.g. MONO0=1, MONO1=0 or MONO2=0 or MONO3=0 is not supported).

50.6.15 ASRC Write Protection Mode Register

Name: ASRC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
`0x535243`	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always read at 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
`0`	The write protection of interrupts is disabled
`1`	The write protection of interrupts is enabled. Any attempt to modify the interrupt configuration is cancelled and leads to an error in the ASRC_WPSR register.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
`0`	The write protection of interrupt registers is disabled
`1`	The write protection of interrupt registers is enabled. Any attempt to modify the interrupt configuration is cancelled and leads to an error in the ASRC_WPSR register.

Bit 0 – WPEN Write Protection Enable

Value	Description
`0`	The write protection is disabled
`1`	The write protection is enabled. All write accesses to configuration registers are canceled and generate an error in the ASRC_WPSR register

50.6.16 ASRC Write Protection Status Register

Name: ASRC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							SWETYP[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE		WPVS
Access					R/W	R/W		R/W
Reset					0	0		0

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A Write-only register has been read.
1	WRITE_RO	A write access has been performed on a Read-only register.
2	UNDEF_RW	Access to an undefined address.

Bits 23:8 – WPSRC[15:0] Write Protection Source (cleared on read)
If a write protection violation has occurred (WPVS=1), this field reports the address of the last violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of ASRC_WPSR.
1	A software error has occurred since the last read of ASRC_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of ASRC_WPSR
1	A peripheral internal sequencer error has occurred since the last read of ASRC_WPSR.

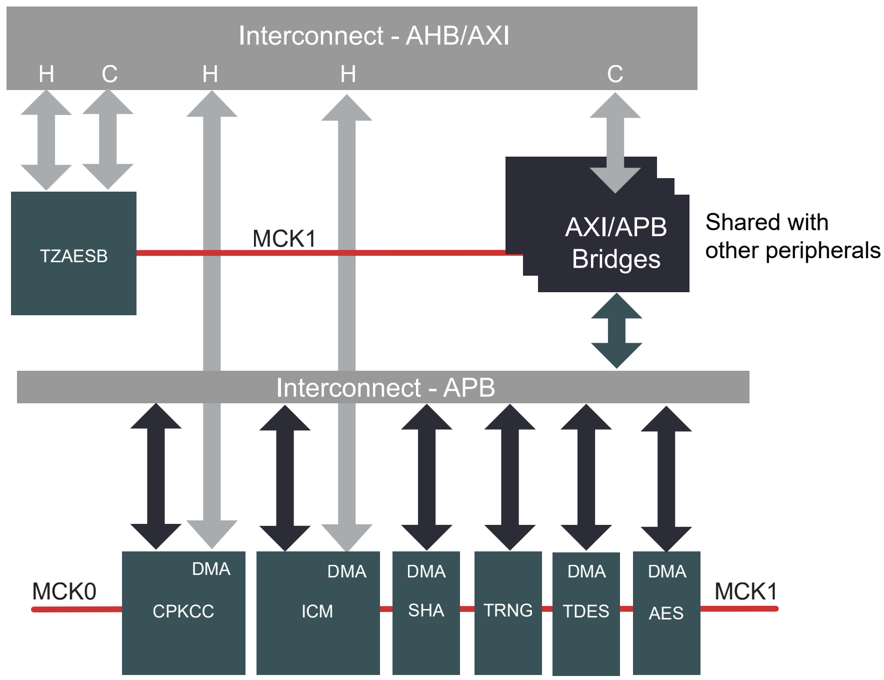
Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of ASRC_WPSR
1	A write protection violation has occurred since the last read of ASRC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported in field WPSRC.

51. CRYPTOGRAPHY SUBSYSTEM

51.1 Block Diagram

Figure 51-1. Cryptography Subsystem Block Diagram



51.2 Components

- Advanced Encryption Standard (AES) engine
- Triple Data Encryption Standard (TDES) engine
- Classical Public Key Cryptography Controller (CPKCC)
- Secure Hash Algorithm (SHA)
- Integrity Check Monitor (ICM)
- TrustZone Advanced Encryption Standard Bridge (TZAESB)
- True Random Number Generator (TRNG)
- One-Time-Programming Memory Controller (OTPC)

Data is moved into and out of the different blocks using a Direct Memory Access (DMA) engine.

51.2.1 Cryptography Subsystem Keybus

The keybus is a private bus transferring keys from hosts (TRNG, OTPC) to clients (AES, TDES, TZAESB, OTPC) with no possibility for processor nor software to read the keys.

All clients except TZAEB have only one physical key, which is either secure or non-secure at time t. TZAESB has two keys, one secure and one non-secure.

Generally, for all IPs:

- Kid = 0, the key is secure.
- Kid = 1, the key is non-secure.

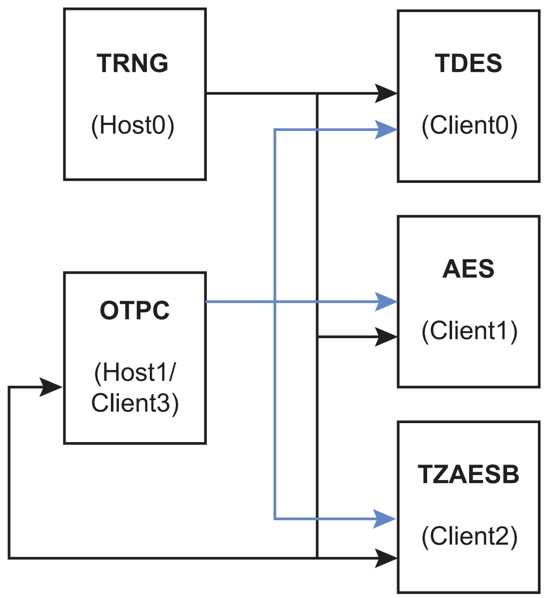
So, if Kid = 1 and the device is in the TrustZone Secure state, access is refused and a violation flag is set in the write protect status of the client.

The key used by the crypto IPs is provided either by the keybus internal register or by the IPs internal key register (KEYWR). To select the keybus as a source for AES, TDES and TZAEB, the PKRS bit must be set in the MR/EMR register.

At host level, the client destination, key length and key type (secure or non-secure) must be defined before starting the transfer.

The device features a keybus system with two hosts and four clients, connected as shown in the following figure.

Figure 51-2. Cryptography Keybus



51.3 Product Dependencies

51.3.1 Clocks

All clocks are controlled by the PMC, which is a part of the system controller.

All peripherals are located on APB Client (APS) matrix, clocked by MCK1, except CPKCC, on APB0 on CPU System and Security (CSS) matrix which is clocked by MCK0.

Note: The MCK0 frequency is directly related to the CPU clock, so any change on the CPU clock impacts MCK0.

51.3.2 Interrupts

Refer to the table Peripheral Identifiers.

51.3.3 Reset

Cryptography peripherals are connected to the processor and peripherals reset line.

51.3.4 I/Os

None.

51.4 Special Functions in SFR/SFRBU

None.

52. TrustZone Advanced Encryption Standard Bridge (TZAESB)

52.1 Description

The TrustZone Advanced Encryption Standard Bridge (TZAESB) provides on-the-fly off-chip memory encryption/decryption compliant with the American *FIPS (Federal Information Processing Standard) Publication 197* specification.

The TZAESB supports one confidentiality mode of operation for symmetrical key block cipher algorithms (CTR), as specified in the *NIST Special Publication 800-38A Recommendation*.

The 128-bit TZAESB key is stored in the TZAESB Key register made of four 32-bit write-only TZAESB Key Word registers (TZAESB_KEYWR0–3). For a software-invisible key transfer, the Private Key bus accesses the Private Key internal register from the TRNG or OTPC. PKRS in the Extended Mode register (TZAESB_EMR) selects either TZAESB_KEYWRx or the Private Key internal register.

The 128-bit initialization vector (if defined) is stored in four write-only 32-bit registers (TZAESB_IVRx).

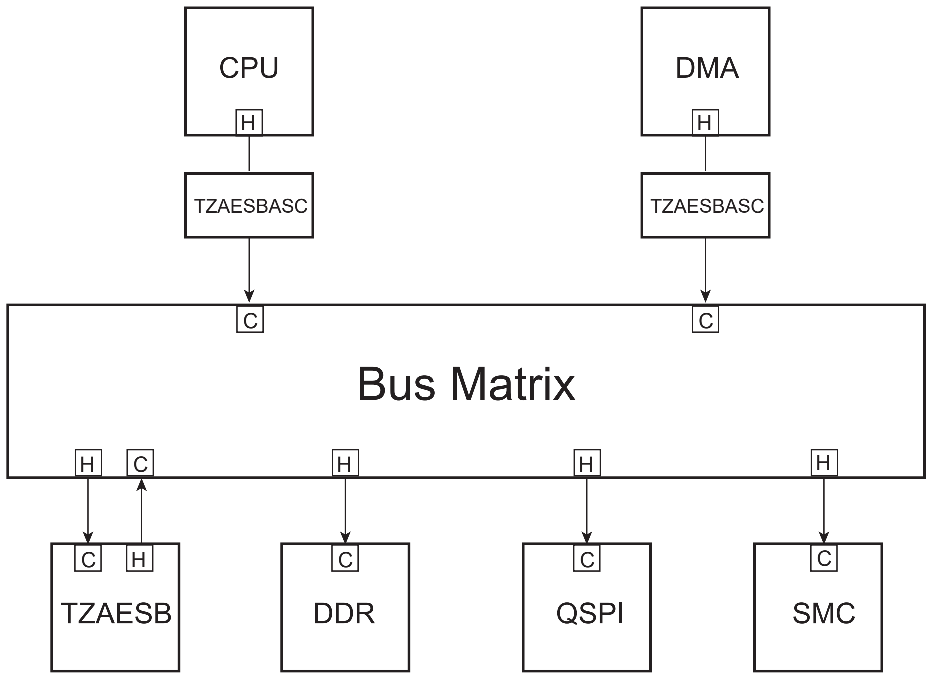
As soon as the initialization vector and the key are configured, the encryption/decryption process may be started.

52.2 Embedded Characteristics

- On-The-Fly Off-Chip Memory Encryption/Decryption
- Two Independent AES Cores with Independent Register Interfaces
- Configurable AES Core Security Attribute
- Compliant with *FIPS Publication 197, Advanced Encryption Standard (AES)*
- 128-bit Cryptographic Key
- 10 Clock Cycles Encryption/Decryption Inherent Processing Time
- Double Input Buffer Optimizes Runtime
- Encryption Mode Based on CTR Mode
- Abnormal Software Access Reports
- Register Write Protection
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software

52.3 Block Diagram

Figure 52-1. TZAESB Block Diagram



52.4 Product Dependencies

52.4.1 Power Management

The TZAESB may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the TZAESB clock.

Note: Enabling TZAESB non-secure clock (PID 98) enables both TZAESB cores.

52.4.2 Interrupt

The TZAESB interface has an interrupt line connected to the Interrupt Controller.

52.5 Functional Description

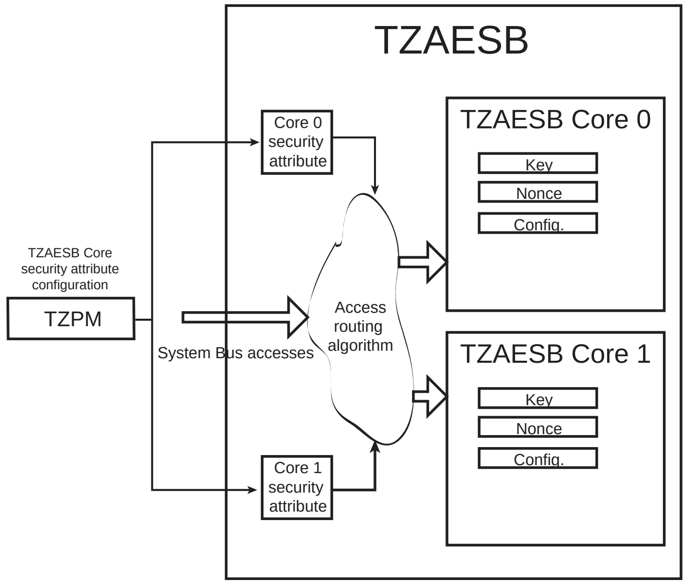
The TrustZone Advanced Encryption Standard Bridge (TZAESB) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The TZAESB algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext.

The TZAESB is capable of using cryptographic keys of 128 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit key is defined in the Key registers (TZAESB_KEYWRx) or in the Private Key internal register that is only writable from the Private Key bus.

The Advanced Encryption Standard Bridge (TZAESB) integrates two AES cores which can be configured separately.

Figure 52-2. TZAESB Top View



Each core has its own configuration and therefore its own cipher key, nonce, etc. Each TZAESB Core configuration (except the security attribute) must be written through the peripheral bus. For details, refer to:

- "TrustZone AES Bridge Address Space Controller (TZAESBASC)" in the section "System Interconnect and Security (SIS)",
- the "Memory Mapping" figure in the section" Memories".

52.5.1 TrustZone Security Attributes

The TZAESB security attribute for each core can be configured to process accesses coming from the secure world on one core and accesses coming from the non-secure world on the other core. Thus, the TZAESB can process two concurrent secure and non-secure data streams. In this configuration, secure accesses are routed to the TZAESB core with a secure attribute and non-secure accesses are routed to the TZAESB core with a non-secure attribute.

Both cores can be configured with the same security attribute to process only one type of access (non-secure or secure), thus doubling encryption performance with respect to standard levels.

The TZAESB core security attribute configuration is done by the TrustZone Peripheral Manager (TZPM) block. Refer to the section "TrustZone Peripheral Manager (TZPM)" for more details.

The TrustZone AES Bridge Address Space Controller (TZAESBASC) defines regions (either secure or non-secure) to the distant memory based on address table definitions. Based on the region definition and the address of the access, the TZAESBASC modifies the secure attribute of the access. Refer to the section "TrustZone AES Bridge Address Space Controller (TZAESBASC)" for more details.

Table 52-1. TZPM Configurations and Expected Behavior

TZAESB Core 1 TZPM Security Bit Configuration	**TZAESB Core 0** TZPM Security Bit Configuration	**Non-secure Access to Core 1 User Interface**	**Non-secure Access to Core 0 User Interface**	**Non-secure Memory Access through TZAESB**	**Secure Memory Access through TZAESB**	**Notes**
Secure	Non-secure	Denied	Accepted	Access to non-secure regions will be accepted and use Core 0, Access to secure regions will be denied.	Accepted, access to secure regions will use core 1 and access to non-secure regions will use Core 0.	Secure world can decrypt both secure and non-secure regions.

...........continued

TZAESB Core 1 TZPM Security Bit Configuration	TZAESB Core 0 TZPM Security Bit Configuration	Non-secure Access to Core 1 User Interface	Non-secure Access to Core 0 User Interface	Non-secure Memory Access through TZAESB	Secure Memory Access through TZAESB	Notes
Secure	Secure	Denied	Denied	Denied	Accepted only if target region is secure, denied otherwise. Accepted access will use the first available core (increased performances)	Both cores must have the same configuration. Performances are increased (both cores can work in parallel).
Non-secure	Non-secure	Accepted	Accepted	Accepted	Accepted	Both cores must have the same configuration. Performances are increased (both cores can work in parallel).
Non-secure	Secure	NA	NA	NA	NA	Forbidden configuration

52.5.2 Automatic Bridge Mode

52.5.2.1 Description

To enter Automatic Bridge mode, TZAESB_MR.OPMOD must be configured to 0x4.

When the TZAESB is connected to the system bus with both Client and Host ports, the Automatic Bridge mode provides automatic encryption/decryption without any action by the user. If TZAESB_MR.OPMOD = 0x4, there is no compliance with the standard CTR mode of operation but the mode of operation is derived from CTR mode.

In case of a write transfer, this mode automatically encrypts the data before writing it to the final client destination. In case of a read transfer, this mode automatically decrypts the data read from the target client before putting it on the system bus.

Therefore, this mode does not work if the automatically encrypted data is moved to another address outside of the TZAESB. Thus for a given data, the encrypted value is not the same if written at different addresses.

52.5.2.2 Configuration

The IV (Initialization Vector) field of the TZAESB Initialization Vector register x (TZAESB_IVRx) can be used to add a nonce in the encryption process in order to add more security (ignored if not filled). In this case, any value encrypted with a given nonce can only be decrypted with this nonce. If another nonce is set for the TZAESB_IVRx.IV, any value encrypted with the previous nonce can no longer be decrypted (see TZAESB_IVRx).

52.5.3 Security Features

52.5.3.1 Private Key Bus

The TZAESB features a Private Key internal register that can be accessed only through the dedicated Private Key bus from the TRNG or OTPC.

The Private Key internal register cannot be read from any peripheral or from software.

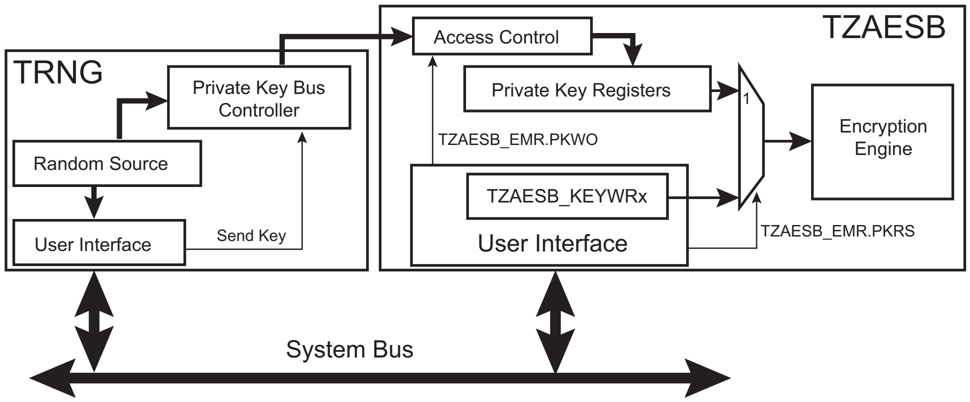
The TZAESB key used by the encryption/decryption engine is either the Private Key internal register content or the TZAESB_KEYWRx registers content.

By default, after a hardware reset, the TZAESB key is provided by the TZAESB_KEYWRx registers. The software can select the Private Key internal register by setting TZAESB_EMR.PKRS. The keys stored in TZAESB_KEYWRx remain available for later use by clearing TZAESB_EMR.PKRS.

Before selecting the Private Key internal register, the software must:

1. Trigger the key transfer over the Private Key bus from the TRNG or OTPC Key Bus host.
2. Wait for completion of the transfer signaled in the Key Bus host Status register.
3. Check for any access violation in TZAESB_WPSR.PKRPVS.

Figure 52-3. Key Selection



52.5.3.2 Unspecified Register Access Detection

When an unspecified register access occurs, TZAESB_ISR.URAD rises. Its source is then reported in TZAESB_ISR.URAT. Only the last unspecified register access is available through TZAESB_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- Mode register written during data processing
- Mode register written during sub-keys generation
- Write-only register read access

TZAESB_ISR.URAD and TZAESB_ISR.URAT can only be reset by TZAESB_CR.SWRST.

52.5.3.3 Clearing Key on Tamper Event

On a tamper detection event, an immediate clear of the scrambling key (TZAESB_KEYWRx) can be performed if TZAESB_MR.TAMPCLR=1.

52.5.3.4 Register Write Protection

To prevent any single software error from corrupting TZAESB behavior, certain registers in the address space can be write-protected by setting WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) in the TZAESB Write Protection Mode register (TZAESB_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the TZAESB Write Protection Status register (TZAESB_WPSR) is set and Write Protection Violation Source (WPVSRC) indicates the register in which the write access has been attempted.

WPVS is automatically cleared after reading TZAESB_WPSR.

The following registers can be write-protected when WPEN is set in TZAESB_WPMR:

- TZAESB Mode Register
- TZAESB Key Word Register x
- TZAESB Initialization Vector Register x
- TZAESB Extended Mode Register

The following registers can be write-protected when WPITEN is set:

- TZAESB Interrupt Enable Register
- TZAESB Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TZAESB Control Register

52.5.3.5 Security and Safety Analysis and Reports

Several types of checks are performed when the TZAESB is enabled.

The peripheral clock of the TZAESB is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TZAESB. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TZAESB_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TZAESB is also monitored and if an abnormal state is detected, the flag TZAESB_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TZAESB are monitored and if an incorrect access is performed, the flag TZAESB_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in TZAESB_WPSR.SWETYP (see TZAESB_WPSR for details). TZAESB_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TZAESB_WPSR is read.

If one of these flags is set, the flag TZAESB_ISR.SECE is set and can trigger an interrupt if TZAESB_IMR.SECE is ‘1’. SECE is cleared by reading TZAESB_ISR.

It is possible to configure an action to be performed by TZAESB as soon as an abnormal event detection occurs. If TZAESB_WPMR.ACTION > 0, either a lock is performed or a lock and immediate clear of the TZAESB_KEYWRx key. If a lock is performed, the current processing is ended normally but no new processing is performed.

A locked state of the TZAESB is unlocked as follows:

1. Read TZAESB_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to TZAESB_CR.UNLOCK.

It is possible to select the type of event that will lock the TZAESB in case of abnormal event detection. See TZAESB_WPMR.ACTION for details.

If TZAESB_MR.TMPCLR=1 and the tamper pin is active, the TZAESB is locked.

52.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TZAESB_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0								START
0x04	TZAESB_MR	31:24	TAMPCLR							
		23:16	CKEY[3:0]							
		15:8		OPMOD[2:0]						
		7:0	PROCDLY[3:0]							
0x08 ... 0x0F	Reserved									
0x10	TZAESB_IER	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								
0x14	TZAESB_IDR	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								
0x18	TZAESB_IMR	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								
0x1C	TZAESB_ISR	31:24								
		23:16					SECE			
		15:8	URAT[3:0]							URAD
		7:0								
0x20	TZAESB_KEYWR0	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x24	TZAESB_KEYWR1	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x28	TZAESB_KEYWR2	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x2C	TZAESB_KEYWR3	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x30 ... 0x5F	Reserved									
0x60	TZAESB_IVR0	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x64	TZAESB_IVR1	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x68	TZAESB_IVR2	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x6C	TZAESB_IVR3	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x70 ... 0xAF	Reserved									
0xB0	TZAESB_EMR	31:24								
		23:16								
		15:8								
		7:0	PKRS	PKWO						
0xB4 ... 0xE3	Reserved									
0xE4	TZAESB_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	TZAESB_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS

52.6.1 TZAESB Control Register

Name: TZAESB_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if WPCREN is cleared in the TZAESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								–

Bit 24 – UNLOCK Unlock Processing
TZAESB_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if TZAESB_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the TZAESB. A software triggered hardware reset of the TZAESB interface is performed.

Bit 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual encryption/decryption process.

52.6.2 TZAESB Mode Register

Name: TZAESB_MR
Offset: 0x04
Reset: 0x00000004
Property: Read/Write

This register can only be written if WPEN is cleared in the TZAESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
	CKEY[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
		OPMOD[2:0]						
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bit	7	6	5	4	3	2	1	0
	PROCDLY[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit 31 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on the TZAESB_KEYWRx key.
1	A tamper detection event immediately clears the TZAESB_KEYWRx key.

Bits 23:20 – CKEY[3:0] Key

Value	Name	Description
0xE	PASSWD	Must be written with 0xE the first time that TZAESB_MR is programmed. For subsequent programming of TZAESB_MR, any value can be written, including that of 0xE. Always reads as 0.

Bits 14:12 – OPMOD[2:0] Operating Mode
If OPMOD is set to 4, there is no compliance with the standard CTR mode of operation but the mode of operation is derived from CTR mode.
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	–	Reserved
1	–	Reserved
2	–	Reserved
3	–	Reserved
4	CTR	Counter mode (16-bit internal counter)

Bits 7:4 – PROCDLY[3:0] Processing Delay
The best performance is achieved with PROCDLY equal to 0.
Processing Time = 12 × (PROCDLY + 1)
The Processing Time represents the number of clock cycles that the TZAESB needs in order to perform one encryption/decryption .

52.6.3 TZAESB Interrupt Enable Register

Name: TZAESB_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if WPITEN is cleared in the TZAESB Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

52.6.4 TZAESB Interrupt Disable Register

Name: TZAESB_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if WPITEN is cleared in the TZAESB Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

52.6.5 TZAESB Interrupt Mask Register

Name: TZAESB_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

52.6.6 TZAESB Interrupt Status Register

Name: TZAESB_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
	URAT[3:0]							URAD
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 19 – SECE Security and/or Safety Event

Value	Description
0	There is no security report in TZAESB_WPSR.
1	One security flag is set in TZAESB_WPSR.

Bits 15:12 – URAT[3:0] Unspecified Register Access
Only the last Unspecified Register Access Type is available through URAT.
URAT is reset only by TZAESB_CR.SWRST.

Value	Name	Description
0	IDR_WR_PROCESSING	Input Data register written during the data processing
1	ODR_RD_PROCESSING	Output Data register read during the data processing
2	MR_WR_PROCESSING	Mode register written during the data processing
3	ODR_RD_SUBKGEN	Output Data register read during the sub-keys generation
4	MR_WR_SUBKGEN	Mode register written during the sub-keys generation
5	WOR_RD_ACCESS	Write-only register read access

Bit 8 – URAD Unspecified Register Access Detection Status
URAD is reset only by TZAESB_CR.SWRST.

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

52.6.7 TZAESB Key Word Register x

Name: TZAESB_KEYWRx
Offset: 0x20 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

TZAESB_KEYWRx registers are not used if the Private Key internal register is selected instead by writing a ‘1’ to TZAESB_EMR.PKRS.
This register can only be written if WPEN is cleared in the TZAESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEYW[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEYW[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEYW[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEYW[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEYW[31:0] Key Word
The four 32-bit Key Word registers set the 128-bit cryptographic key used for encryption/decryption.
TZAESB_KEYWR0 corresponds to the first word of the key, TZAESB_KEYWR3 to the last one.
These registers are write-only to prevent the key from being read by another application.

52.6.8 TZAESB Initialization Vector Register x

Name: TZAESB_IVRx
Offset: 0x60 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

These registers are not used in ECB mode and must not be written. For Automatic Bridge mode, the IV input value corresponds to the initial nonce.
This register can only be written if WPEN is cleared in the TZAESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector
The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.
TZAESB_IVR0 corresponds to the first word of the Initialization Vector, TZAESB_IVR3 to the last one.
These registers are write-only to prevent the Initialization Vector from being read by another application.
For CBC mode, the IV input value corresponds to the initialization vector.
For CTR mode, the IV input value corresponds to the initial counter value.

52.6.9 TZAESB Extended Mode Register

Name: TZAESB_EMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPEN is cleared in the TZAESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PKRS	PKWO						
Access	R/W	R/W						
Reset	0	0						

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The key used by the TZAESB is in the TZAESB_KEYWRx.
1	The key used by the TZAESB is in the Private Key internal register written through the Private Key bus.

Bit 6 – PKWO Private Key Write Once
Once PKWO is set to '1', only a hardware reset sets this bit to ‘0’ internally. Writing it to '0' with a register access has no impact (although the field will be read to value '0').

Value	Description
0	The Private Key internal register can be written multiple times through the Private Key bus.
1	The Private Key internal register can be written only once through the Private Key bus until hardware reset.

52.6.10 TZAESB Write Protection Mode Register

Name: TZAESB_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414553	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags is set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the TZAESB_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the TZAESB_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the TZAESB_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the TZAESB_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued. Moreover, the TZAESB_KEYWRx key is immediately cleared.

Value	Name	Description
5	CLEAR_CGD_SEQE	If a processing is in progress when the TZAESB_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued. Moreover, the TZAESB_KEYWRx key is immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the TZAESB_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TZAESB_CR.UNLOCK command is issued. Moreover, the TZAESB_KEYWRx key is immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TZAESB_WPSR.WPVSRC and the last software control error type is reported in TZAESB_WPSR.SWETYP. The TZAESB_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TZAESB_WPSR.WPVSRC and only the first software control error type is reported in TZAESB_WPSR.SWETYP. The TZAESB_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

52.6.11 TZAESB Write Protection Status Register

Name: TZAESB_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality
1	ERROR	An access is performed into key, input data, control registers while the TZAESB is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	TZAESB is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of TZAESB_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A Private Key bus access, key write, init value write, output data read or TZAESB_MR and TZAESB_EMR write has been performed while a process is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the key is not fully loaded into the TZAESB_KEYWRx registers.

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Internal Register Protection Violation Status (cleared on read)

Value	Description
0	No Private Key internal register access violation has occurred since the last read of TZAESB_WPSR.

Value	Description
1	A Private Key internal register access violation has occurred since the last read of TZAESB_WPSR.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TZAESB_WPSR.
1	A software error has occurred since the last read of TZAESB_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TZAESB_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TZAESB_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of TZAESB_WPSR. Under normal operating conditions, this bit is always cleared.
1	A clock glitch has occurred since the last read of TZAESB_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of TZAESB_WPSR.
1	A write protect violation has occurred since the last read of TZAESB_WPSR. The address offset of the violated register is reported into field WPVSRC.

53. TrustZone AES Bridge Address Space Controller (TZAESBASC)

53.1 Description

The TrustZone Advanced Encryption Standard Bridge Address Space Controller (TZAESBASC) is used to define which part of the product memory space is accessed through the TrustZone AES Bridge.

The user can define up to 8 regions. Each region can be sized independently with a granularity of 4 Kbytes, and is defined by a base address and a top address.

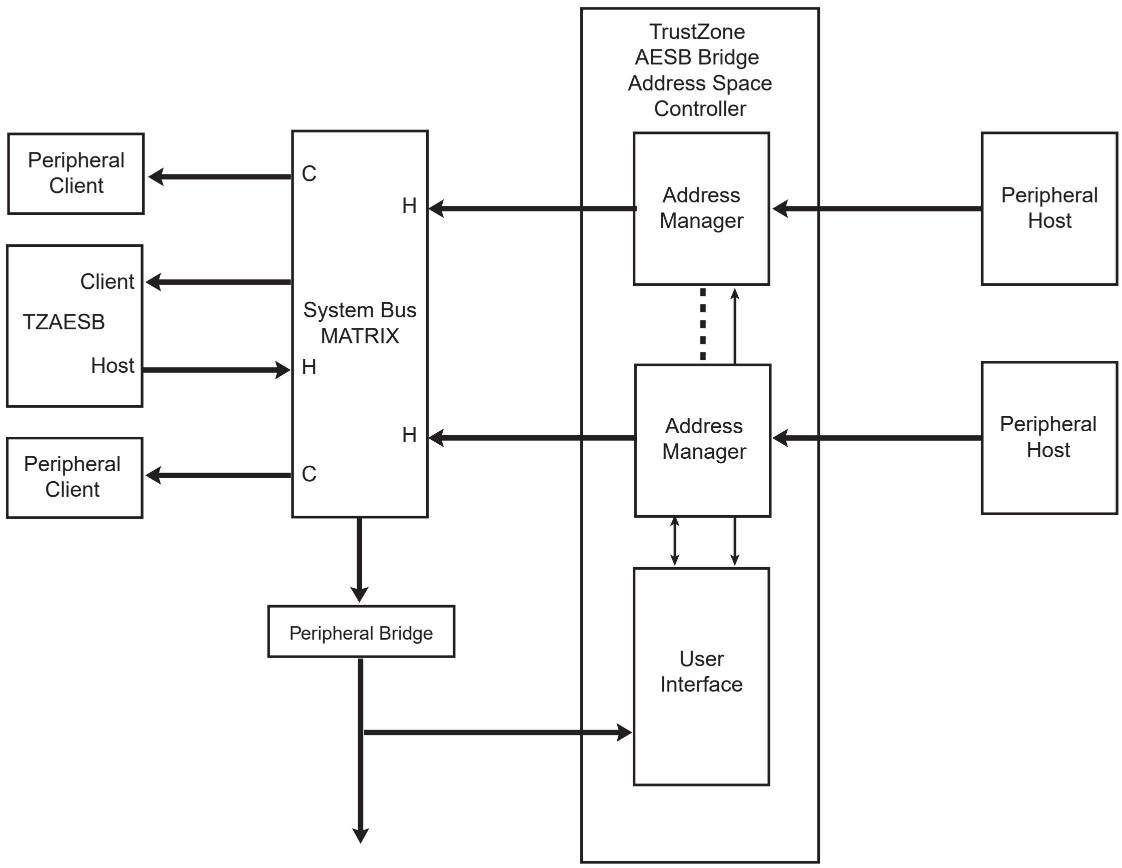
The address bus of each peripheral host which can access the TrustZone AES Bridge goes through a TZAESBASC Address Manager. If a transfer address matches an enabled region, the address is modified in order to route the transferred data through the TrustZone AES Bridge, otherwise the transferred data is routed directly to/from the addressed peripheral client.

53.2 Embedded Characteristics

- Up to 8 Regions
- Independently Sizeable Regions with 4-Kbyte Granularity
- Each Region defined by a Base Address and a Top Address
- Address Error if Base Address and Top Address are Inconsistent, or if Programmed Addresses cannot be reached by the TrustZone AES Bridge

53.3 Block Diagram

Figure 53-1. TZAESBASC Block Diagram

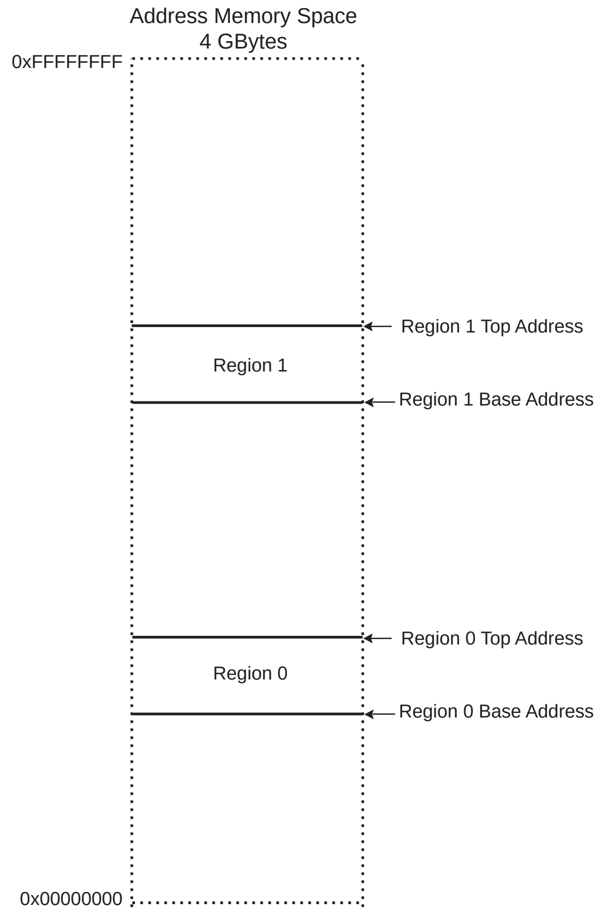


53.4 Functional Description

53.4.1 Operating Modes

To define a region in the address memory space which will be accessed through the TrustZone AES Bridge, the user must write the base address in the Region Base Address register and the top address in the Region Top Address register (see TZAESBASC_RBARx and TZAESBASC_RTARx). Bits [11:0] of these registers are discarded and always read at 0 because the size granularity of a region is 4 Kbytes.

Figure 53-2. Address Memory Space Regions



Then, the user must define the security attribute of the region by writing the Region Security register (see TZAESBASC_RSECR). If the region is "non-secure" and is accessible by both the "secure world" and the "normal world", then the corresponding SEC bit must be written to 1. If the region is "secure" and is accessible by the "secure-world" only, then the SEC bit must be written to 0. The programmed security attribute must match the one programmed in the MMU (Memory Management Unit) of the CPU.

Once the Region Address and Region Security registers are written, the region can be enabled by writing 1 in the corresponding bit of the Region Enable register (see TZAESBASC_RER). To ensure address consistency, only one region can be enabled at a time. Writing more than one bit at 1 in TZAESBASC_RER has no effect.

The region is disabled by writing 1 in the corresponding bit of the Region Disable register (see TZAESBASC_RDR). Disabling a region resets all related information of this region. The corresponding memory areas are automatically cleared.

Because the synchronization of the TZAESBASC Address Managers can take time, the enable/disable of the region may not be immediate. The corresponding Enable Status (ES) bit in the Region Status register (see TZAESBASC_RSR) must be polled to wait for the effective enable/disable of the region. The configuration status can also be checked by reading the Region Synchronization Status register (see TZAESBASC_RSSR).

If a region base address and top address are not consistent (i.e., if the top address is lower than or equal to the base address), if a region memory space overlaps another enabled region, or if the region memory space is not reachable by the TrustZone AES bridge, the corresponding AER error bit in the Region Error Status register is set to 1 (see TZAESBASC_RESR). Bit AER is cleared after the read of TZAESBASC_RESR.

⚠ CAUTION — Configuration must not be done during accesses. It may lead to unpredictable behavior.

53.4.2 Write Protection Registers

To prevent any single software error from corrupting TZAESBASC behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the TZAESBASC Write Protection Mode register (TZAESBASC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the TZAESBASC Write Protection Status register (TZAESBASC_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading TZAESBASC_WPSR.

The following registers can be write-protected by setting TZAESBASC_WPMR.WPEN:

- TZAESBASC_RBARx
- TZAESBASC_RTARx
- TZAESBASC_RSECR
- TZAESBASC_RER
- TZAESBASC_RDR

53.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TZAESBASC_RBAR0	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x04	TZAESBASC_RTAR0	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x08	TZAESBASC_RBAR1	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x0C	TZAESBASC_RTAR1	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x10	TZAESBASC_RBAR2	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x14	TZAESBASC_RTAR2	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x18	TZAESBASC_RBAR3	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x1C	TZAESBASC_RTAR3	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x20	TZAESBASC_RBAR4	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x24	TZAESBASC_RTAR4	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x28	TZAESBASC_RBAR5	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x2C	TZAESBASC_RTAR5	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x30	TZAESBASC_RBAR6	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x34	TZAESBASC_RTAR6	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	TZAESBASC_RBAR7	31:24	BASE[31:24]							
		23:16	BASE[23:16]							
		15:8	BASE[15:8]							
		7:0	BASE[7:0]							
0x3C	TZAESBASC_RTAR7	31:24	TOP[31:24]							
		23:16	TOP[23:16]							
		15:8	TOP[15:8]							
		7:0	TOP[7:0]							
0x40 ... 0x7F	Reserved									
0x80	TZAESBASC_RSECR	31:24								
		23:16								
		15:8								
		7:0	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
0x84	TZAESBASC_RER	31:24								
		23:16								
		15:8								
		7:0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
0x88	TZAESBASC_RDR	31:24								
		23:16								
		15:8								
		7:0	DIS7	DIS6	DIS5	DIS4	DIS3	DIS2	DIS1	DIS0
0x8C	TZAESBASC_RSR	31:24								
		23:16								
		15:8								
		7:0	ES7	ES6	ES5	ES4	ES3	ES2	ES1	ES0
0x90	TZAESBASC_RESR	31:24								
		23:16								
		15:8								
		7:0	AER7	AER6	AER5	AER4	AER3	AER2	AER1	AER0
0x94	TZAESBASC_RSSR	31:24								
		23:16								
		15:8								
		7:0								SYNC
0x98 ... 0xE3	Reserved									
0xE4	TZAESBASC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	TZAESBASC_WPSR	31:24								
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0								WPVS

53.5.1 TZAESBASC Region x Base Address Register

Name: TZAESBASC_RBARx
Offset: 0x00 + x*0x08 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BASE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BASE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BASE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASE[31:0] Region x Base Address
Base Address bits [11:0] are always 0 because a region size granularity is 4 Kbytes.

53.5.2 TZAESBASC Region x Top Address Register

Name: TZAESBASC_RTARx
Offset: 0x04 + x*0x08 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TOP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TOP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TOP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TOP[31:0] Region x Top Address
Top Address bits [11:0] are always 0 because a region size granularity is 4 Kbytes.

53.5.3 TZAESBASC Region Security Register

Name: TZAESBASC_RSECR
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – SECx Region x Security

Value	Description
0	Region x is secure.
1	Region x is non-secure.

53.5.4 TZAESBASC Region Enable Register

Name: TZAESBASC_RER
Offset: 0x84
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ENx Region x Enable

Value	Description
0	No effect.
1	Enables the corresponding region.

53.5.5 TZAESBASC Region Disable Register

Name: TZAESBASC_RDR
Offset: 0x88
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DIS7	DIS6	DIS5	DIS4	DIS3	DIS2	DIS1	DIS0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – DISx Region x Disable

Value	Description
0	No effect.
1	Disables the corresponding region.

53.5.6 TZAESBASC Region Status Register

Name: TZAESBASC_RSR
Offset: 0x8C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ES7	ES6	ES5	ES4	ES3	ES2	ES1	ES0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ESx Region x Enable Status

Value	Description
0	The corresponding region is disabled.
1	The corresponding region is enabled.

53.5.7 TZAESBASC Region Error Status Register

Name: TZAESBASC_RESR
Offset: 0x90
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	AER7	AER6	AER5	AER4	AER3	AER2	AER1	AER0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – AERx Region x Address Error

Value	Description
0	No region x address error has occurred.
1	A region x address error has occurred since the last read of the Region Error Status register.

53.5.8 TZAESBASC Region Synchronization Status Register

Name: TZAESBASC_RSSR
Offset: 0x94
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SYNC
Access								R
Reset								0

Bit 0 – SYNC Region Synchronization Status

Value	Description
0	Synchronization of the region configuration is done.
1	A region configuration is in progress.

53.5.9 TZAESBASC Write Protection Mode Register

Name: TZAESBASC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
`0x415343`	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
`0`	Disables the write protection if WPKEY corresponds to 0x415343 (“ASC” in ASCII).
`1`	Enables the write protection. All accesses to configuration registers are canceled and generate an error in the TZAESBASC_WPSR register.

53.5.10 TZAESBASC Write Protection Status Register

Name: TZAESBASC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

See 53.4.2. Write Protection Registers for the list of write-protected registers.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of TZAESBASC_WPSR.
1	A write protection violation has occurred since the last read of TZAESBASC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

54. Advanced Encryption Standard (AES)

54.1 Description

The Advanced Encryption Standard (AES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 197 specification.

The AES supports the following confidentiality modes of operation for symmetrical key block cipher algorithms: ECB, CBC, OFB, CFB, CTR and XTS, as specified in the NIST Special Publication 800-38A Recommendation and NIST Special Publication 800-38E Recommendation, as well as Galois/Counter Mode (GCM) as specified in the NIST Special Publication 800-38D Recommendation. It is compatible with all these modes via DMA Controller channels, minimizing processor intervention for large buffer transfers.

The AES key can be either loaded by the software or loaded in an invisible manner from the software.

The 128-bit/192-bit/256-bit AES key is stored in the AES Key register made of four/six/eight 32-bit write-only AES Key Word registers (AES_KEYWR0–7). For a software-invisible key transfer, the Private Key Bus accesses the Private Key Internal Register from the TRNG or OTPC. The bit PKRS in the Extended Mode register (AES_EMR) selects either AES_KEYWRx or the Private Key Internal Register.

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit write-only AES Input Data registers (AES_IDATAR0–3) and AES Initialization Vector registers (AES_IVR0–3).

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data are ready to be read out on the four 32-bit AES Output Data registers (AES_ODATAR0–3) or through the DMA channels.

54.2 Embedded Characteristics

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128-bit/192-bit/256-bit Cryptographic Key
- 10/12/14 Clock Cycles Encryption/Decryption Inherent Processing Time with a 128-bit/192-bit/256-bit Cryptographic Key
- Double Input Buffer Optimizes Runtime
- Automatic Padding supported for IPSec and SSL standards
- IPSec and SSL Protocol Layers Improved Performances (Tightly coupled with SHA)
- Support of the Modes of Operation Specified in the NIST Special Publication 800-38A and NIST Special Publication 800-38D and NIST Special Publication 800-38E:
  – Electronic Codebook (ECB)
  – Cipher Block Chaining (CBC) including CBC-MAC
  – Cipher Feedback (CFB)
  – Output Feedback (OFB)
  – Counter (CTR)
  – Galois/Counter Mode (GCM)
  – XEX-Based Tweaked-Codebook Mode (XTS)
- 8, 16, 32, 64 and 128-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Abnormal Software Access and Internal Sequencer Integrity Check Reports
- Register Write Protection
- Temporary Secure Storage for Keys
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software
- Connection to DMA Optimizes Data Transfers for all Operating Modes

54.3 Product Dependencies

54.3.1 Power Management

The AES is clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the AES clock.

54.3.2 Interrupt Sources

The AES interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by the Interrupt Mask register (AES_IMR) and the Interrupt Status register (AES_ISR), whereas the interrupt line for safety/security functions is driven by Write Protection Status register (AES_WPSR) flags. If one of the flags AES_WPSR.WPVS, AES_WPSR.CGD, AES_WPSR.SEQE or AES_WPSR.SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the AES.

54.4 Functional Description

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AES Mode register (AES_MR) allows selection between the encryption and the decryption processes.

The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit/192-bit/256-bit key is defined in the user interface AES_KEYWRx register or in the Private Key Internal Register that is only writable from the Private Key Bus.

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in AES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. AES_IVRx are also used by the CTR mode to set the counter value.

54.4.1 AES Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The AES interface requires little-endian format words. However, in accordance with the protocol of the FIPS 197 specification, data is collected, processed and stored by the AES algorithm in big-endian form.

The following example illustrates how to configure the AES:

If the first 64 bits of a message (according to FIPS 197, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then AES_IDATAR0 and AES_IDATAR1 registers must be written with the following pattern:

- AES_IDATAR0 = 0xcadefeca
- AES_IDATAR1 = 0x67452301

54.4.2 Operating Modes

The AES supports the following modes of operation:

- ECB: Electronic Codebook
- CBC: Cipher Block Chaining
  – CBC-MAC: Useful for CMAC hardware acceleration
- OFB: Output Feedback
- CFB: Cipher Feedback
  – CFB8 (CFB where the length of the data segment is 8 bits)
  – CFB16 (CFB where the length of the data segment is 16 bits)
  – CFB32 (CFB where the length of the data segment is 32 bits)

  - CFB64 (CFB where the length of the data segment is 64 bits)
  - CFB128 (CFB where the length of the data segment is 128 bits)
- CTR: Counter
- GCM: Galois/Counter Mode
- XTS: XEX-based Tweaked-codebook Mode

Data pre-processing, data post-processing and data chaining for the concerned modes are performed automatically. Refer to the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D* for more complete information.

Mode selection is done by configuring AES_MR.OPMOD.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

In CFB mode, five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of AES_MR.CFBS.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 Mbyte of data. If the file to be processed is greater than 1 Mbyte, this file must be split into fragments of 1 Mbyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AES_IDATARx, AES_IVRx must be fully programmed with the initial counter value. For any fragment, after the transfer is completed and prior to transferring the next fragment, AES_IVRx must be programmed with the appropriate counter value.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 Mbyte, the size of the first fragment to be processed must be 1 Mbyte minus 16 × (initial value) to prevent a rollover of the internal 16-bit counter.

To have a sequential increment, the counter value must be programmed with the value programmed for the previous fragment + 2^{16} (or less for the first fragment).

All AES_IVRx fields must be programmed to take into account the possible carry propagation.

54.4.3 Last Output Data Mode (CBC-MAC)

This mode is used to generate cryptographic checksums on data (MAC) by means of cipher block chaining encryption algorithm (CBC-MAC algorithm for example).

The CMAC algorithm is a variant of CBC-MAC with post-processing requiring one-block encryption in ECB mode. Thus CBC-MAC is useful to accelerate CMAC.

After each end of encryption/decryption, the output data are available either on AES_ODATARx for Manual and Auto mode, or at the address specified in the receive buffer pointer for DMA mode (see the table Last Output Data Mode Behavior versus Start Modes).

AES_MR.LOD allows retrieval of only the last data of several encryption/decryption processes.

Therefore, there is no need to define a read buffer in DMA mode.

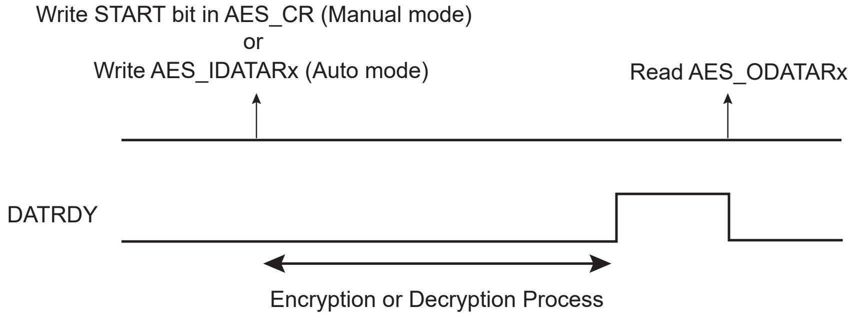
This data are only available in AES_ODATARx.

54.4.3.1 Manual and Auto Modes

54.4.3.1.1 If AES_MR.LOD = 0

The DATRDY flag is cleared when at least one AES_ODATARx is read (see the following figure).

Figure 54-1. Manual and Auto Modes with AES_MR.LOD = 0



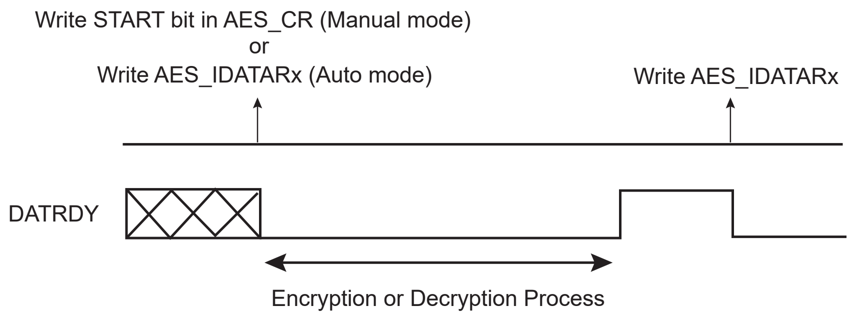
If the user does not want to read AES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user cannot know the end of the following encryptions/decryptions.

54.4.3.1.2 If AES_MR.LOD = 1

This mode is optimized to process AES CBC-MAC operating mode.

The DATRDY flag is cleared when at least one AES_IDATAR is written (see the following figure). No additional AES_ODATAR reads are necessary between consecutive encryptions/decryptions.

Figure 54-2. Manual and Auto Modes with AES_MR.LOD = 1



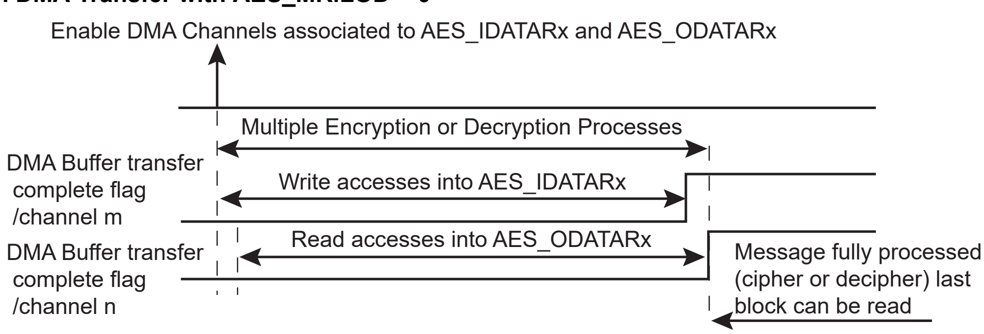
54.4.3.2 DMA Mode

54.4.3.2.1 If AES_MR.LOD = 0

This mode may be used for all AES operating modes except CBC-MAC where AES_MR.LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to AES_ODATARx (see the following figure). Two DMA channels are required: one for writing message blocks to AES_IDATARx and one to obtain the result from AES_ODATARx.

Figure 54-3. DMA Transfer with AES_MR.LOD = 0



54.4.3.2.2 If AES_MR.LOD = 1

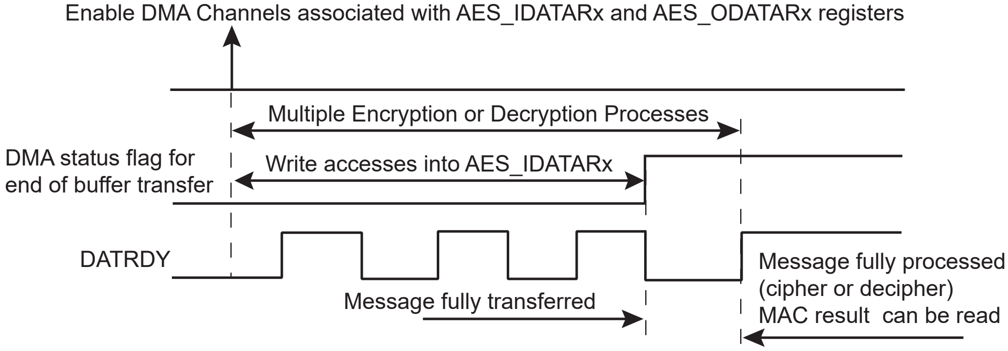
This mode is optimized to process AES CBC-MAC operating mode.

The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see the following figure).

The DMA receive channel must not be used. Prior to reading the CBC-MAC result, AES_MR.SMOD must be written to ‘0’. To restart a CBC-MAC on a new buffer, AES_MR.SMOD must be written to ‘2’.

The output data are only available on AES_ODATARx.

Figure 54-4. DMA Transfer with AES_MR.LOD = 1



The following table summarizes the different cases.

Table 54-1. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	AES_MR.LOD = 0	AES_MR.LOD = 1	AES_MR.LOD = 0	AES_MR.LOD = 1
DATRDY Flag Clearing Condition(1)	At least one AES_ODATAR must be read	At least one AES_IDATAR must be written	Not used	Managed by the DMA
End of Encryption/ Decryption Notification	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then AES DATRDY flag
Encrypted/ Decrypted Data Result Location	In AES_ODATARx	In AES_ODATARx	At the address specified in the Channel Buffer Transfer Descriptor	In AES_ODATARx

Note:

1. Depending on the mode, there are other ways of clearing the DATRDY flag. See AES_ISR.

⚠WARNING In DMA mode, reading AES_ODATARx before the last data transfer may lead to unpredictable results.

54.4.4 Galois/Counter Mode (GCM)

54.4.4.1 Description

GCM comprises the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag (the AES CTR engine and the GHASH engine are depicted in the following figure.

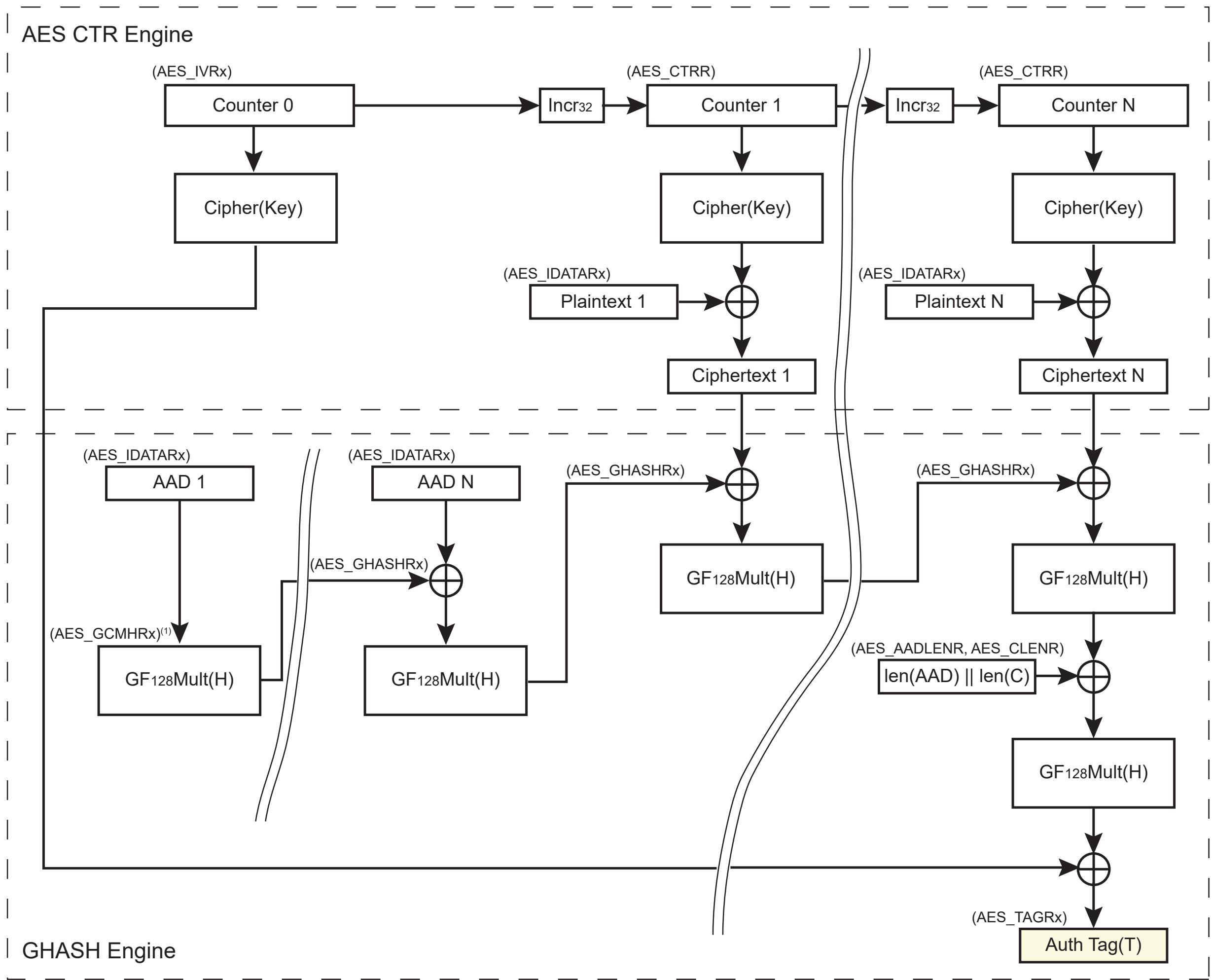
The GHASH engine processes data packets after the AES operation. GCM assures the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. GCM can also provide assurance of data that is not encrypted. Refer to *NIST Special Publication 800-38D* for more complete information.

GCM can be used with or without the DMA host. Messages may be processed as a single complete packet of data or they may be broken into multiple packets of data over time.

GCM processing is computed on 128-bit input data fields. There is no support for unaligned data. The AES key length can be whatever length is supported by the AES module.

The recommended programming procedure when using DMA is described in the section GCM Processing.

Figure 54-5. GCM Block Diagram



Note: 1. Optional

54.4.4.2 Key Writing and Automatic Hash Subkey Calculation

Whenever a new key is written to the hardware, two automatic actions are processed:

- GCM Hash Subkey *H* generation—The GCM hash subkey (*H*) is automatically generated. The GCM hash subkey generation must be complete before doing any other action. AES_ISR.DATRDY indicates when the subkey generation is complete (with interrupt if configured). The GCM hash subkey calculation is processed with the formula *H* = CIPHER(Key, <128 bits to zero>). The generated GCM *H* value is then available in AES_GCMHRx. If the application software requires a specific hash subkey, the automatically generated *H* value can be overwritten in AES_GCMHRx.
  AES_GCMHRx can be written after the end of the hash subkey generation (see AES_ISR.DATRDY) and prior to starting the input data feed.
- AES_GHASHRx Clear—AES_GHASHRx are automatically cleared. If a hash initial value is needed for the GHASH, it must be written to AES_GHASHRx
  - after writing AES_KEYWRx, if any
  - before starting the input data feed

54.4.4.3 GCM Processing

GCM processing is made up of three phases:

1. Processing the Additional Authenticated Data (*AAD*), hash computation only.
2. Processing the Ciphertext (C), hash computation + ciphering/deciphering.
3. Generating the Tag using length of *AAD*, length of C and J_0 (refer to NIST documentation for details).

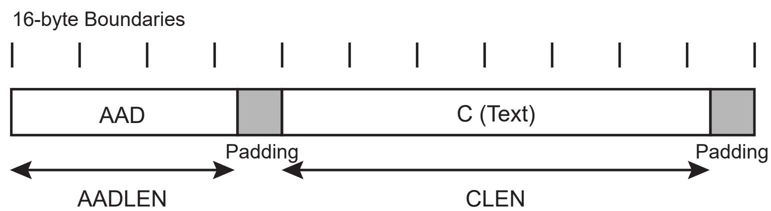
The Tag generation can be done either automatically, after the end of *AAD*/C processing if AES_MR.GTAGEN is set, or manually using AES_GHASHRx.GHASH (see subsections Processing a Complete Message with Tag Generation and Manual GCM Tag Generation for details).

54.4.4.3.1 Processing a Complete Message with Tag Generation

Use this procedure only if J0 four LSB bytes ≠ 0xFFFFFFFF.

Note: If J_0 four LSB bytes = 0xFFFFFFFF or if the value is unknown, use the procedure described in Processing a Complete Message without Tag Generation followed by the procedure in Manual GCM Tag Generation.

Figure 54-6. Full Message Alignment



To process a complete message with Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to ‘1’.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation J_0 = IV || 0^{31} || 1 when len(IV) = 96 and J_0 = $GHASH_H$(IV || 0^{s+64} || [len(IV)]64) if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Wait for TAGRDY to be set (use interrupt if needed), then read AES_TAGRx.TAG to obtain the authentication tag of the message.

54.4.4.3.2 Processing a Complete Message without Tag Generation

Processing a message without generating the Tag can be used to customize the Tag generation, or to process a fragmented message. To manually generate the GCM Tag, see Manual GCM Tag Generation.

To process a complete message without Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to ‘0’.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J0 value as described in NIST documentation J_0 = IV || 0^{31} || 1 when len(IV) = 96 and J_0 = $GHASH_H$(IV || 0^{s+64} || [len(IV)]64) if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).

7. Make sure the last output data have been read if AES_CLENR.CLEN ≠ 0 (or wait for DATRDY), then read AES_GHASHRx.GHASH to obtain the hash value after the last processed data.

54.4.4.3.3 Processing a Fragmented Message without Tag Generation

If needed, a message can be processed by fragments, in such case automatic GCM Tag generation is not supported.

To process a message by fragments, the sequence is as follows:

- First fragment:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation J_0 = IV || 0^{31} || 1 when len(IV) = 96 and J_0 = $GHASH_H$(IV || 0^{s+64} || [len(IV)]64) if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the first fragment, or set the fields with the full message length (both configurations work).
6. Fill AES_IDATARx.IDATA with the first fragment of the message to process (aligned on 16-byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in AAD phase).

- Next fragment (or last fragment):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Set AES_IVRx.IV as follows:
   – If the first block of the fragment is a block of Additional Authenticated data, set AES_IVRx.IV with the J_0 initial value
   – If the first block of the fragment is a block of Plaintext data, set AES_IVRx.IV with a value constructed as follows: 'LSB96(J_0) || CTR' value, (96 bit LSB of J_0 concatenated with saved CTR value from previous fragment).
4. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the current fragment, or set the fields with the remaining message length, both configurations work.
5. Fill AES_GHASHRx.GHASH with the value stored after the previous fragment.
6. Fill AES_IDATARx.IDATA with the current fragment of the message to process (aligned on 16 byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in AAD phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

Note: Step 1 and 2 are required only if the value of the concerned registers has been modified.

Once the last fragment has been processed, the GHASH value will allow manual generation of the GCM tag. See Manual GCM Tag Generation.

54.4.4.3.4 Manual GCM Tag Generation

This section describes the last steps of the GCM Tag generation.

The Manual GCM Tag Generation is used to complete the GCM Tag Generation when the message has been processed without Tag Generation.

Note: The Message Processing without Tag Generation must be finished before processing the Manual GCM Tag Generation.

To generate a GCM Tag manually, the sequence is as follows:

Processing S = $GHASH_H$ (*AAD* || 0v || C || 0u || [len(*AAD*)]64 || [len(C)]64):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN to 0x10 (16 bytes) and AES_CLENR.CLEN to '0'. This will allow running a single $GHASH_H$ on a 16-byte input data (see the following figure).
4. Fill AES_GHASHRx.GHASH with the state of the GHASH field stored at the end of the message processing.
5. Fill AES_IDATARx.IDATA according to the SMOD configuration used with 'len(*AAD*)64 || len(*C*)64' value as described in the NIST documentation and wait for DATRDY to be set; use interrupt if needed.
6. Read AES_GHASHRx.GHASH to obtain the current value of the hash.

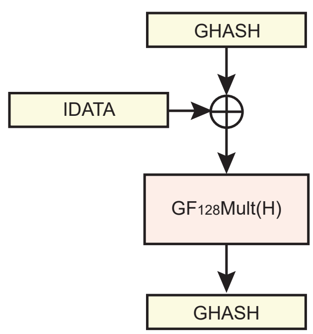
Processing T = GCTRK(J_0, S):

1. Set AES_MR.OPMOD to CTR.
2. Set AES_IVRx.IV with 'J_0' value.
3. Fill AES_IDATARx.IDATA with the GHASH value read at step 6 and wait for DATRDY to be set (use interrupt if needed).
4. Read AES_ODATARx.ODATA to obtain the GCM Tag value.

Note: Step 4 is optional if the GHASH field is to be filled with value '0' (0 length packet for instance).

54.4.4.3.5 Processing a Message with only AAD (GHASHH)

Figure 54-7. Single $GHASH_H$ Block Diagram (AADLEN ≤ 0x10 and CLEN = 0)



It is possible to process a message with only *AAD* setting the CLEN field to '0' in AES_CLENR, this can be used for J0 generation when len(*IV*) ≠ 96 for instance.

Example: Processing J_0 when len(*IV*) ≠ 96

To process J_0 = $GHASH_H$(*IV* || 0^{s+64} || [len(*IV*)]64), the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write AES_KEYWRx and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN with 'len(*IV* || 0^{s+64} || [len(*IV*)]64)' in and AES_CLENR.CLEN to '0'. This will allow running a $GHASH_H$ only.
4. Fill AES_IDATARx.IDATA with the message to process (*IV* || 0^{s+64} || [len(*IV*)]64) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a $GHASH_H$ step is over (use interrupt if needed).
5. Read AES_GHASHRx.GHASH to obtain the J_0 value.
Note: The GHASH value can be overwritten at any time by writing the value of AES_GHASHRx.GHASH, used to perform a $GHASH_H$ with an initial value for GHASH (write GHASH field between step 3 and step 4 in this case).

54.4.4.3.6 Processing a Single GF128 Multiplication

The AES can also be used to process a single multiplication in the Galois field on 128 bits (GF_{128}) using a single $GHASH_H$ with custom *H* value (see the figure above).

To run a GF_{128} multiplication (A x B), the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to ‘0’.

1. Configure AES_AADLENR.AADLEN with 0x10 (16 bytes) and AES_CLENR.CLEN to ‘0’. This will allow running a single $GHASH_H$.
2. Fill AES_GCMHRx.H with B value.
3. Fill AES_IDATARx.IDATA with the A value according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASHH computation is over (use interrupt if needed).
4. Read AES_GHASHRx.GHASH to obtain the result.

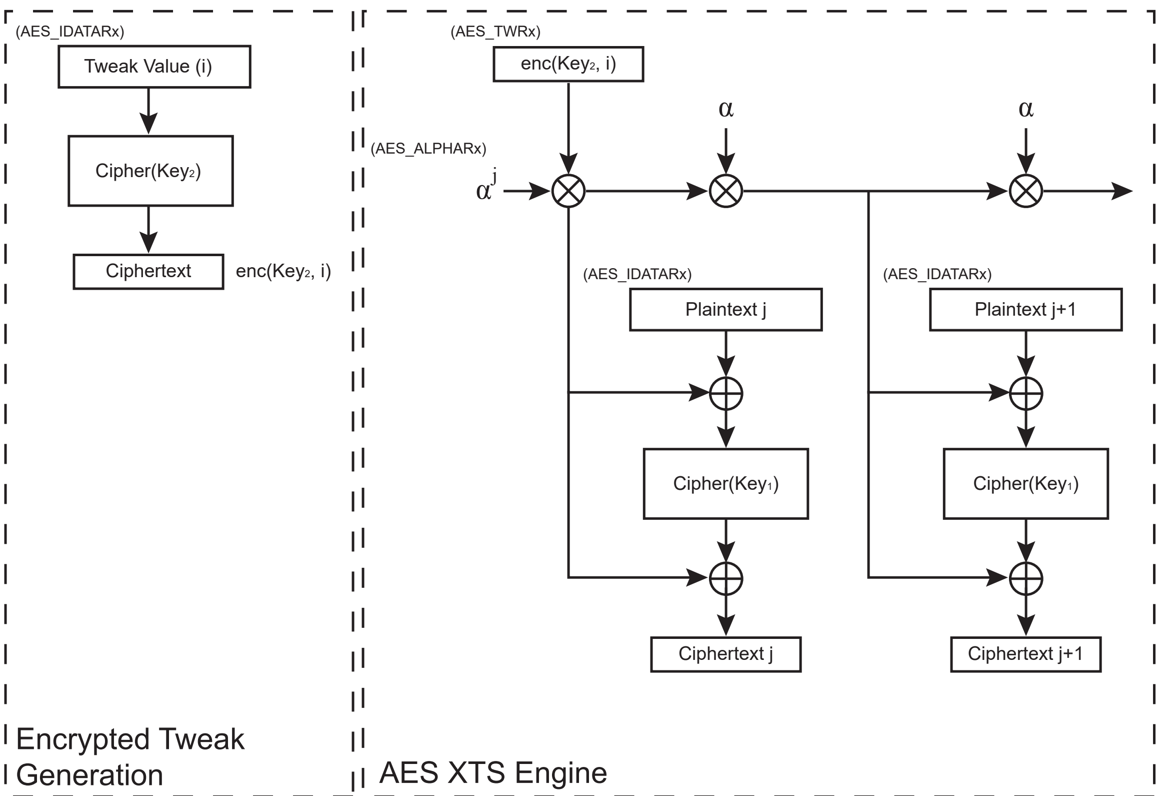
Note: AES_GHASHRx.GHASH can be initialized with a value C between step 3 and step 4 to run a ((A XOR C) x B) GF_{128} multiplication.

54.4.5 XEX-based Tweaked-codebook Mode (XTS)

XTS mode comprises the AES engine with XOR on inputs and outputs. After each encryption/decryption, the value used for the XOR is multiplied by the first $GF(2^{128})$ alpha primitive (0x2) and then used for the next encryption/decryption. The XTS mode uses two different keys and defines a Tweak Value (i) as additional input.

XTS processing is computed on 128-bit input data fields. There is no support for unaligned data (padding must be done manually if needed). The AES key length can be any length supported by the AES module.

Figure 54-8. XTS Block Diagram



54.4.5.1 XTS Processing Procedure

XTS processing comprises two phases:

1. Generate encrypted tweak with Key2 (this step is only required for the first processing, further consecutive processing does not require this step).
2. Process the data giving encrypted tweak and first alpha primitive for the first encryption/decryption.

54.4.5.1.1 Encrypted Tweak Generation

In the case of a new encryption/decryption, it is necessary to first encrypt the Tweak Value (i) with Key2. Here are the steps to follow to perform this step:

1. Set AES_MR.OPMODE to ECB and AES_MR.CIPHER to '1'.
2. Write the Key2.
3. Fill AES_IDATARx.IDATA with the Tweak value (i) according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES_ODATARx.

54.4.5.1.2 Data Processing

To process data using XTS mode, follow the steps below:

1. Set AES_MR.OPMODE to XTS.
2. Write the Key1.
3. Only if the data to process is the first to be processed in the data unit, or if the data block to process is not consecutive to the previous processed data block in the same data unit, then two additional mandatory steps are required:
   a. AES_TWRx must be written with the encrypted Tweak Value (see Encrypted Tweak Generation for details) with bytes swapped as described in AES Register Endianness.

   b. Write AES_ALPHARx with the alpha primitive corresponding to the block number in the data unit.
4. Fill AES_IDATARx.IDATA with the data to process according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES_ODATARx. Repeat Step 4 as long as consecutive data blocks are processed in the same data unit.

54.4.6 Double Input Buffer

AES_IDATARx can be double-buffered to reduce the runtime of large files.

This mode allows a new message block to be written when the previous message block is being processed. This is only possible when DMA accesses are performed (AES_MR.SMOD = 2).

AES_MR.DUALBUFF must be set to '1' to access the double buffer.

54.4.7 Temporary Secured Storage for Keys

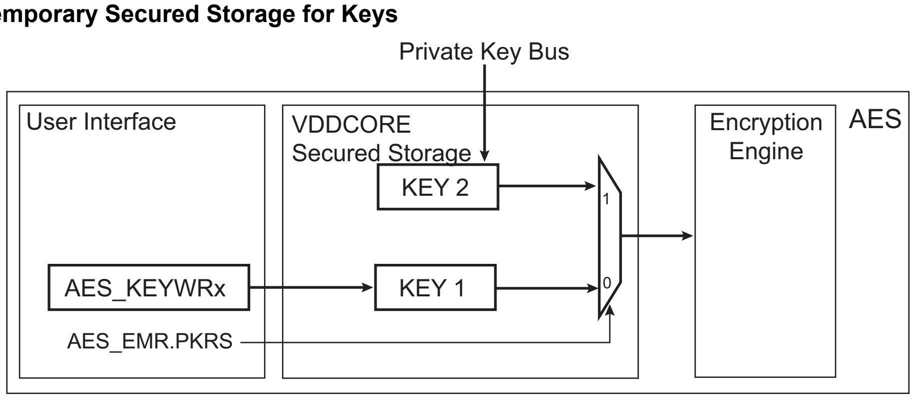
The AES provides secure storage for up to 256-bit keys. The storage is available while VDDCORE voltage is supplied.

The keys can be only written in AES internal registers and are not readable. Moreover, the internal registers holding the keys are buried in the overall product logic area during the physical implementation.

One key can be loaded by software by writing the Key Word registers (AES_KEYWRx).

One key can be loaded by Private Key bus only.

Figure 54-9. Temporary Secured Storage for Keys



54.4.8 Start Modes

AES_MR.SMOD allows selection of the encryption (or decryption) Start mode.

54.4.8.1 Manual Mode

The sequence of actions is as follows:

1. Write AES_MR with all required fields, including but not limited to SMOD and OPMOD. (Write AES_EMR.PKRS according to the type of key to be loaded).
2. Write the 128-bit/192-bit/256-bit AES key in AES_KEYWRx or in the Private Key internal registers.
3. Write the initialization vector (or counter) in AES_IVRx.
   Note: AES_IVRx concerns all modes except ECB.
4. Set the bit DATRDY (Data Ready) in the AES Interrupt Enable register (AES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized AES_IDATARx (see the following table).
6. Set the START bit in the AES Control register (AES_CR) to begin the encryption or the decryption process.
7. When processing completes, the DATRDY flag in the AES Interrupt Status register (AES_ISR) is raised. If an interrupt has been enabled by setting AES_IER.DATRDY, the interrupt line of the AES is activated.
8. When software reads one of AES_ODATARx, AES_IER.DATRDY is automatically cleared.

Table 54-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	AES_IDATAR0 and AES_IDATAR1
32-bit CFB	AES_IDATAR0
16-bit CFB	AES_IDATAR0
8-bit CFB	AES_IDATAR0
CTR	All
GCM	All
XTS	All

Notes:
1. In 64-bit CFB mode, writing to AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.
2. In 32, 16, and 8-bit CFB modes, writing to AES_IDATAR1, AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

54.4.8.2 Auto Mode

The Auto Mode is similar to the manual one, except that in this mode, as soon as the correct number of AES_IDATARx is written, processing is automatically started without any action in AES_CR.

54.4.8.3 DMA Mode

The DMA Controller can be used in association with the AES to perform an encryption/decryption of a buffer without any action by software during processing.

AES_MR.SMOD must be configured to 2 and the DMA must be configured with non-incremental addresses.

For all operating modes except CBC-MAC (AES_MR.LOD=1), 2 DMA channels must be programmed (transmit and receive). In CBC-MAC, only 1 transmit channel must be programmed.

The start address of any transfer descriptor must be configured with the address of AES_IDATAR0.

The DMA chunk size configuration depends on the AES mode of operation and is summarized in the following table.

When writing data to AES with a first DMA channel, data are first fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the AES with the second DMA channel, the source data is the data read from AES and data destination is the memory buffer. In this case, the source data size depends on the AES mode of operation, as shown in the following table.

Table 54-3. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	4	Word
CBC	4	Word
OFB	4	Word
CFB 128-bit	4	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte
CTR	4	Word
GCM	4	Word
XTS	4	Word

54.4.9 Automatic Padding Mode

When Automatic Padding mode is configured, the message is automatically padded after the last block is written. Depending on the size of the message, either a padding is performed after the last part of the message and padding blocks are added, or only padding blocks are added.

IPSec and SSL padding standards are both supported.

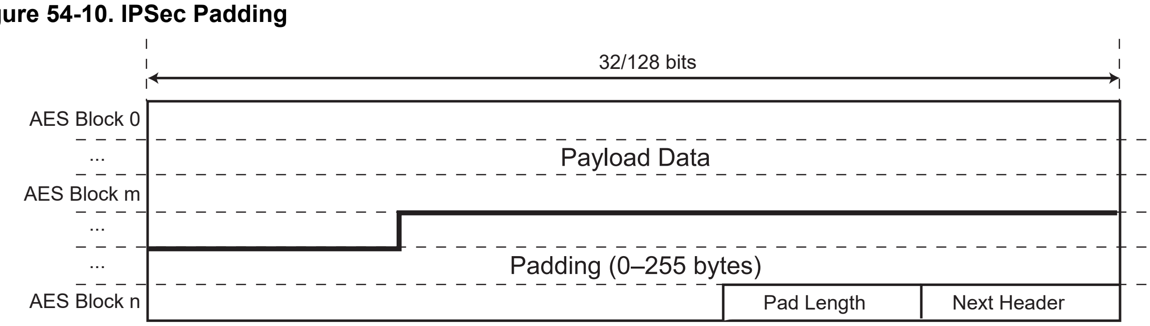
The auto padding feature only supports CBC and CTR modes.

Note: When automatic padding is enabled and AES_MR.SMOD=2, AES_MR.DUALBUFF must be cleared.

54.4.9.1 IPSec Padding

Automatic Padding is enabled by writing a ‘1’ to AES_EMR.APEN. IPSEC padding mode is selected by writing a ‘0’ to AES_EMR.APM.

Figure 54-10. IPSec Padding



Each byte of the padding area contains incremental integer values.

The “Pad Length” in bytes is configured in AES_EMR.PADLEN and the “Next Header” value is configured in AES_EMR.NHEAD. AES_EMR.PADLEN must be configured with the length of the padding section, not including the length of the “Pad Length” and “Next Header” sections.

The BCNT field in the AES Byte Counter register (AES_BCNT) defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the sum of the length of the message (Payload Data) and of the length of the Padding, Pad Length (1 byte) and Next Header (1 byte) sections is a multiple of the AES block size (128 bits).

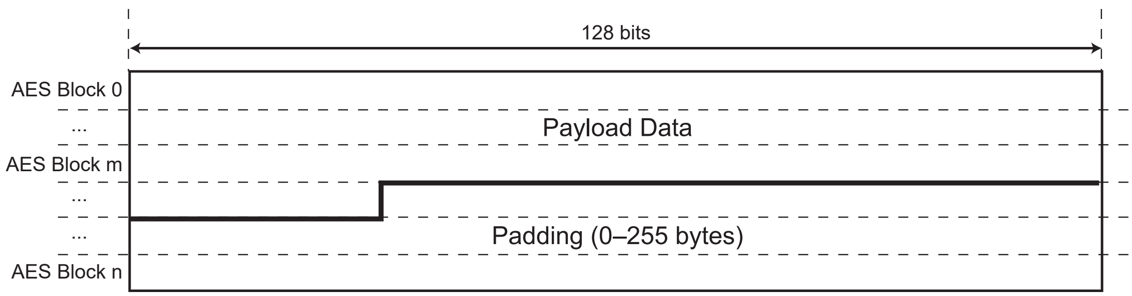
To process an IPSec message using auto-padding, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to ‘1’, AES_EMR.APM to ‘0’, AES_EMR.PADLEN to the desired padding length in byte and AES_EMR.NHEAD to the desired Next Header field value.
3. Configure AES_BCNT.BCNT with the whole message length, without padding, in byte.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block, write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

54.4.9.2 SSL Padding

Auto Padding is enabled by writing a ‘1’ to AES_EMR.APEN and SSL padding mode is selected by writing a ‘1’ to AES_EMR.APM.

Figure 54-11. SSL Padding



Each byte of the padding area contains the padding length.

The padding length is configured in AES_EMR.PADLEN.

AES_BCNT.BCNT defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the length of the message plus the length of the padding section is a multiple of the AES block size (128 bits).

To process a complete SSL message, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to ‘1’, AES_EMR.APM to ‘1’, AES_EMR.PADLEN to the desired padding length in bytes.
3. Set AES_BCNT.BCNT with the whole message length, without padding, in bytes.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

54.4.9.3 Flags

AES_ISR.EOPAD rises as soon as the automatic padding phase is over, meaning that all the extra padding blocks have been processed. Reading AES_ISR clears this flag.

AES_ISR.PLENERR indicates an error in the frame configuration, meaning that the whole message length including padding does not respect the standard selected. AES_ISR.PLENERR rises at the end of the frame in case of wrong message length and is cleared reading AES_ ISR.

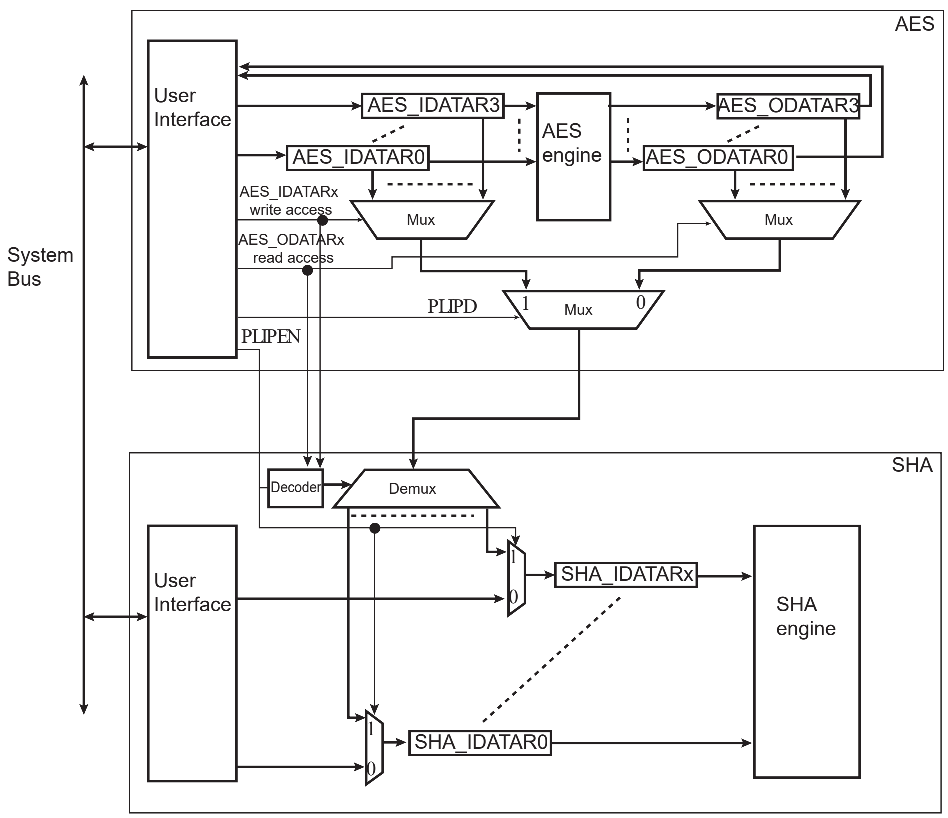
In IPSec/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

54.4.10 Secure Protocol Layers Improved Performances

Secure protocol layers such as IPSec require encryption and authentication. For IPSec, the authentication is based on HMAC, thus SHA is required. To optimize performance, the AES embeds a mode of operation, used with DMA only, that enables the SHA module to process the input or output data of the AES module. If this mode is enabled, write access is required only into AES_IDATARx registers, since SHA input data registers are automatically written by AES without software intervention. When the DMA is configured to transfer a buffer of data (input frame), only one transfer descriptor is required for both authentication and encryption/decryption processes and only one buffer is transferred through the system bus (reducing the load of the system bus).

Improved performance for secure protocol layers requires AES_EMR.PLIPEN to be set.

Figure 54-12. Secure Protocol Layers Improved Performances Block Diagram



54.4.10.1 Cipher Mode

When AES_EMR.PLIPD is cleared and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is first encrypted with the AES module and the encrypted message is authenticated with the SHA module. Therefore, when AES_EMR.PLIPD is cleared, AES_ODATARx are selected and sent to SHA_IDATARx as soon as AES_ODATARx are read. A read access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_ODATARx values into the corresponding SHA_IDATARx without software intervention.

54.4.10.2 Decipher Mode

When AES_EMR.PLIPD is written to ‘1’ and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is decrypted with the AES module and also sent to SHA for authentication. Therefore, when AES_EMR.PLIPD=1, AES_IDATARx are selected and sent to SHA_IDATARx as soon as AES_IDATARx are written. A write access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_IDATARx values into the corresponding SHA_IDATARx without software intervention.

54.4.10.3 Encapsulating Security Payload (ESP) IPSec Examples

The following examples describe how to configure AES and SHA to optimize processing an ESP IPSec frame for maximum performance.

The cipher (or decipher) of an ESP IPSec frame requires both encryption (or decryption) and authentication.

For cipher, the input frame located in the system memory must first be padded and the resulting buffer encrypted. The encrypted frame must be written back to the system memory and sent to the authentication module.

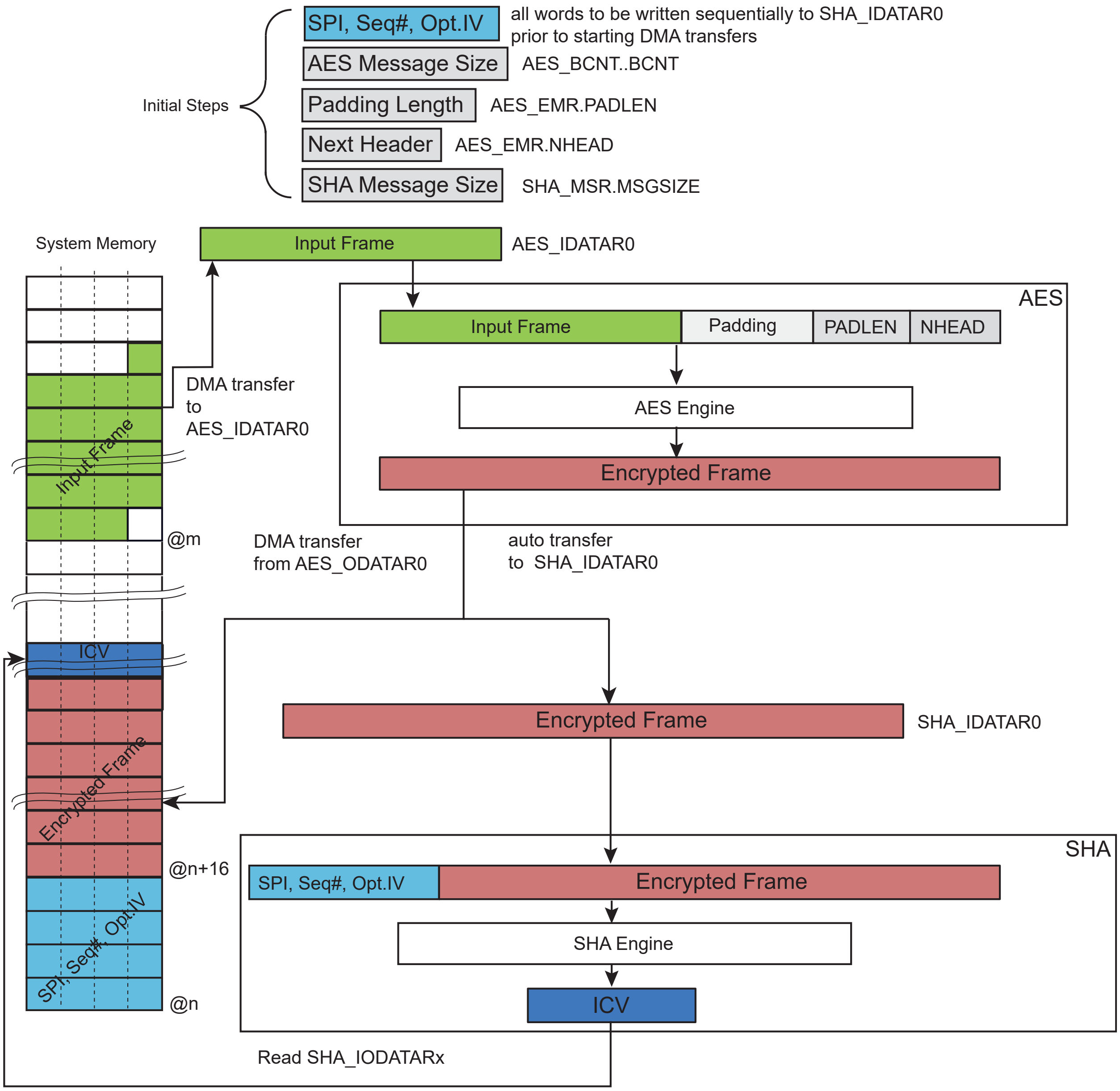
When the AES module is configured to improve the performance of the secure protocol layers (AES_EMR.PLIPEN = 1), the data transfers are simplified, limiting the bandwidth requirements on the system bus.

Before configuring the DMA to start the transfer of the data buffer (input frame) to the AES, the following actions must be taken in registers:

- AES_BCNT.BCNT must be configured with the length of the message (Input Frame).
- The padding length of the AES must be configured in AES_EMR.PADLEN. See Automatic Padding Mode to configure Automatic Padding mode.
- The next header value must be configured in AES_EMR.NHEAD.
- AES_MR.SMOD and SHA_MR.SMOD must be configured to 2.
  Note: When automatic padding is enabled and AES_MR.SMOD = 2 , AES_MR.DUALBUFF must be cleared.
- The SHA_MSR.MSGSIZE must be configured with the length of the authentication message including the optional extended sequence number (ESN) and header and trailer information required by the authentication algorithm used (HMAC, etc.). Refer to the section “Secure Hash Algorithm (SHA)” for more details on configuration for optimized processing of header information.
- The Security Parameter Index (SPI, sequence number (SEQ#)) and the optional Initialization Vector (IV) must be configured sequentially in SHA_IDATAR0.
- A first DMA transfer descriptor must be configured to transfer the input frame from the system memory to the AES input data registers (AES_IDATARx), and a second DMA descriptor must be configured to transfer the encrypted frame from AES to the system memory.
  Note: If AES_EMR.PLIPEN = 1 , there is no need to define a transfer descriptor to load the encrypted frame into the SHA input data registers because the transfer is automatically performed while the second descriptor transfer is in progress.

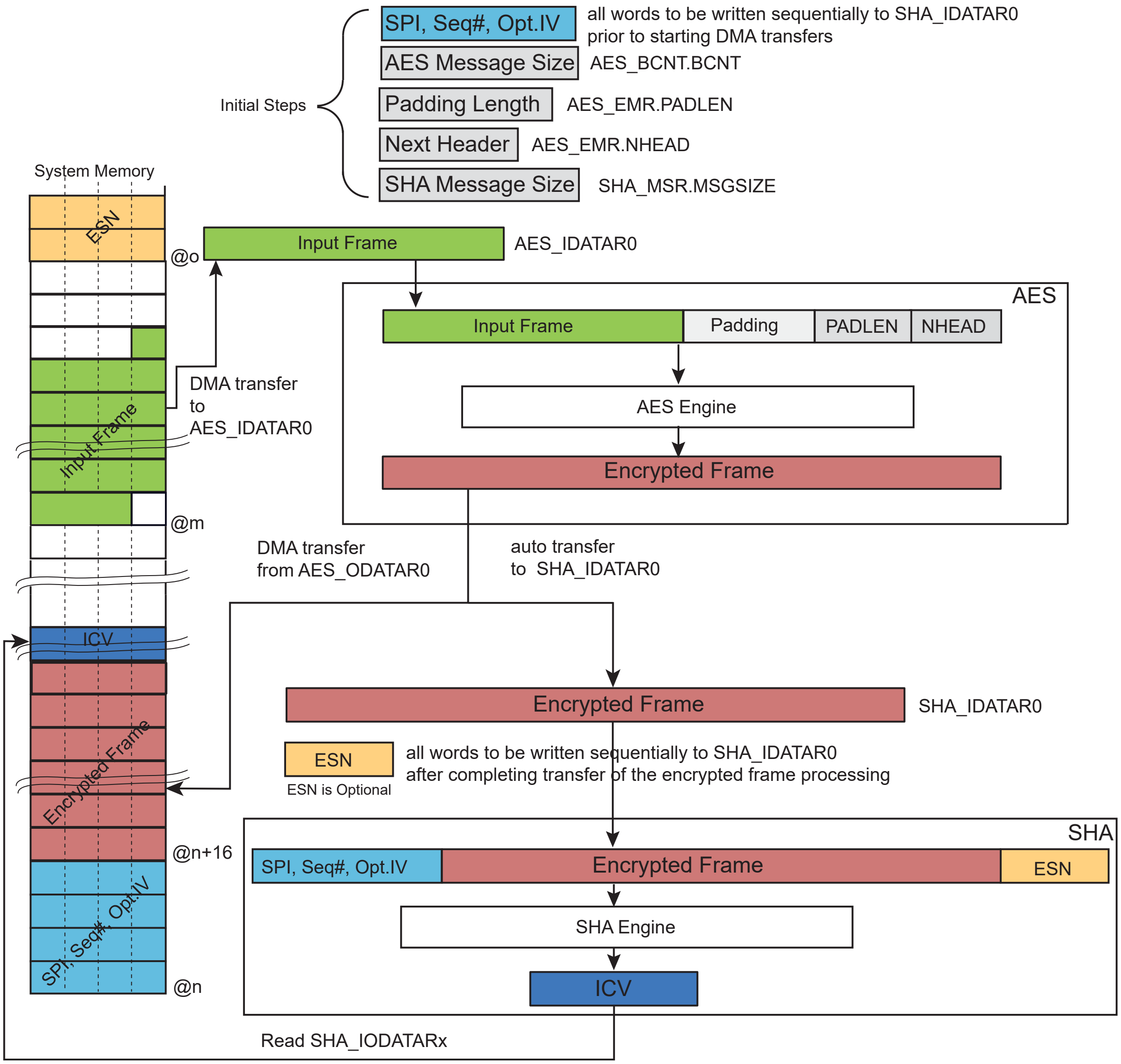
See the following figures.

Figure 54-13. Generation of an ESP IPSec Frame without ESN



If the optional extended sequence number is required for authentication, wait for the AES-to-system memory DMA buffer transfer to complete before configuring the ESN value. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register. See the following figure.

Figure 54-14. Generation of an ESP IPSec Frame with ESN



To decipher an ESP IPSec frame without the optional ESN trailer information, two DMA channels are required and the SHA must be configured in Automatic padding mode.

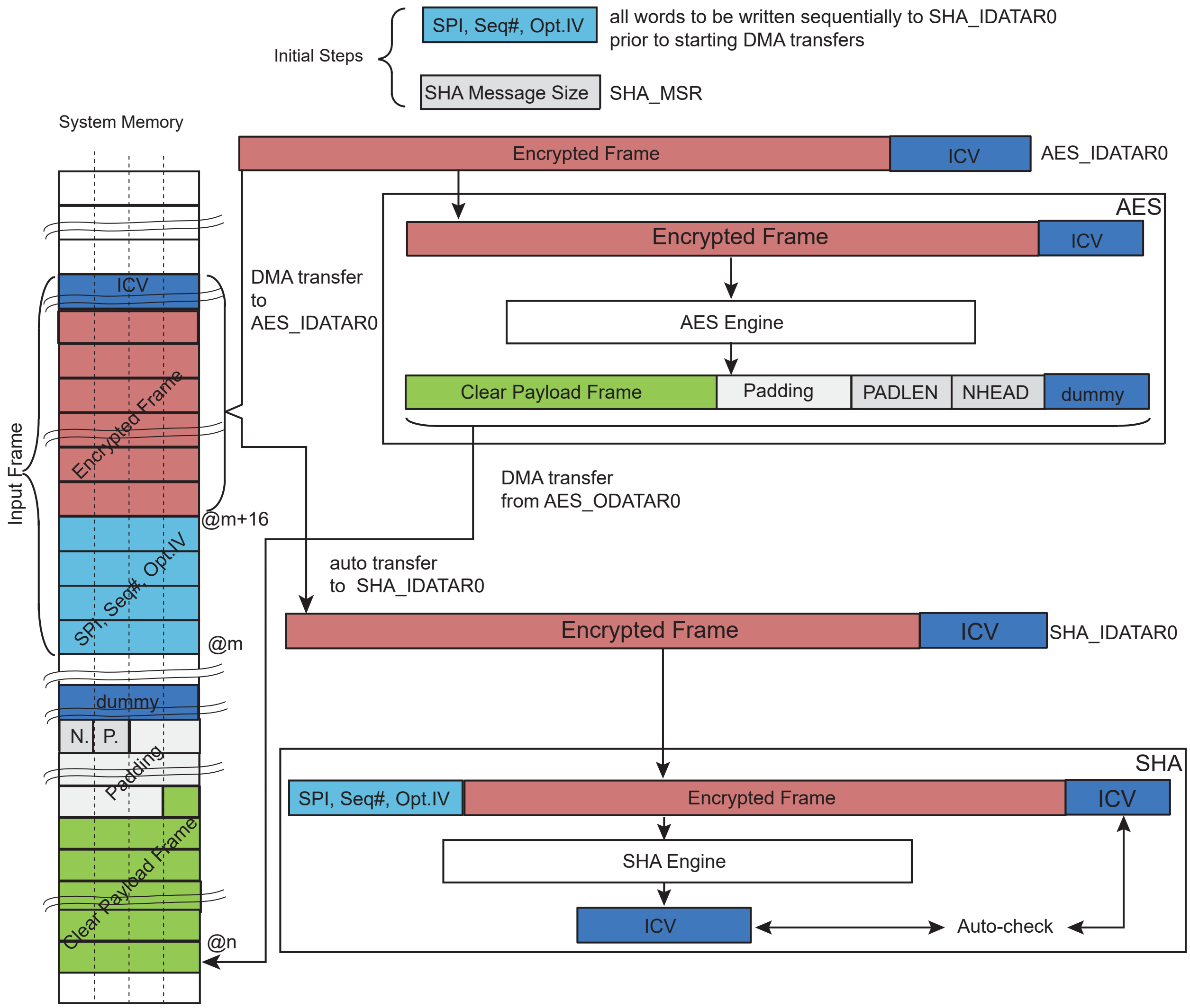
Note: AES automatic padding must be disabled when deciphering a frame.

- A first DMA transfer descriptor must be configured to load the received encrypted frame from the system memory to AES_IDATARx for decryption. The start address of the first transfer descriptor must be defined after the SPI, SEQ#, and optional IV (see the following figure).
- A second DMA descriptor must be configured to transfer the decrypted frame from AES_ODATARx to the system memory.
- AES_EMR.PLIPEN and AES_EMR.PLIPD must be written to '1' so that the data buffer is written in AES_IDATARx and in SHA_IDATARx.

The SHA has the capability to perform an automatic check with an expected integrity check value if this value is appended at the end of the frame buffer (SHA_MR.CHECK=2). Thus, if the first transfer descriptor includes the ICV for SHA, the first DMA transfer allows the decryption and authentication processes including the automatic check. The decrypted part resulting from ICV is not required for downstream processing and must be considered as dummy data.

The end of the decryption and authentication processes occur when flag SHA_ISR.CHECKF=1. The authentication status is provided by SHA_ISR.CHKST.

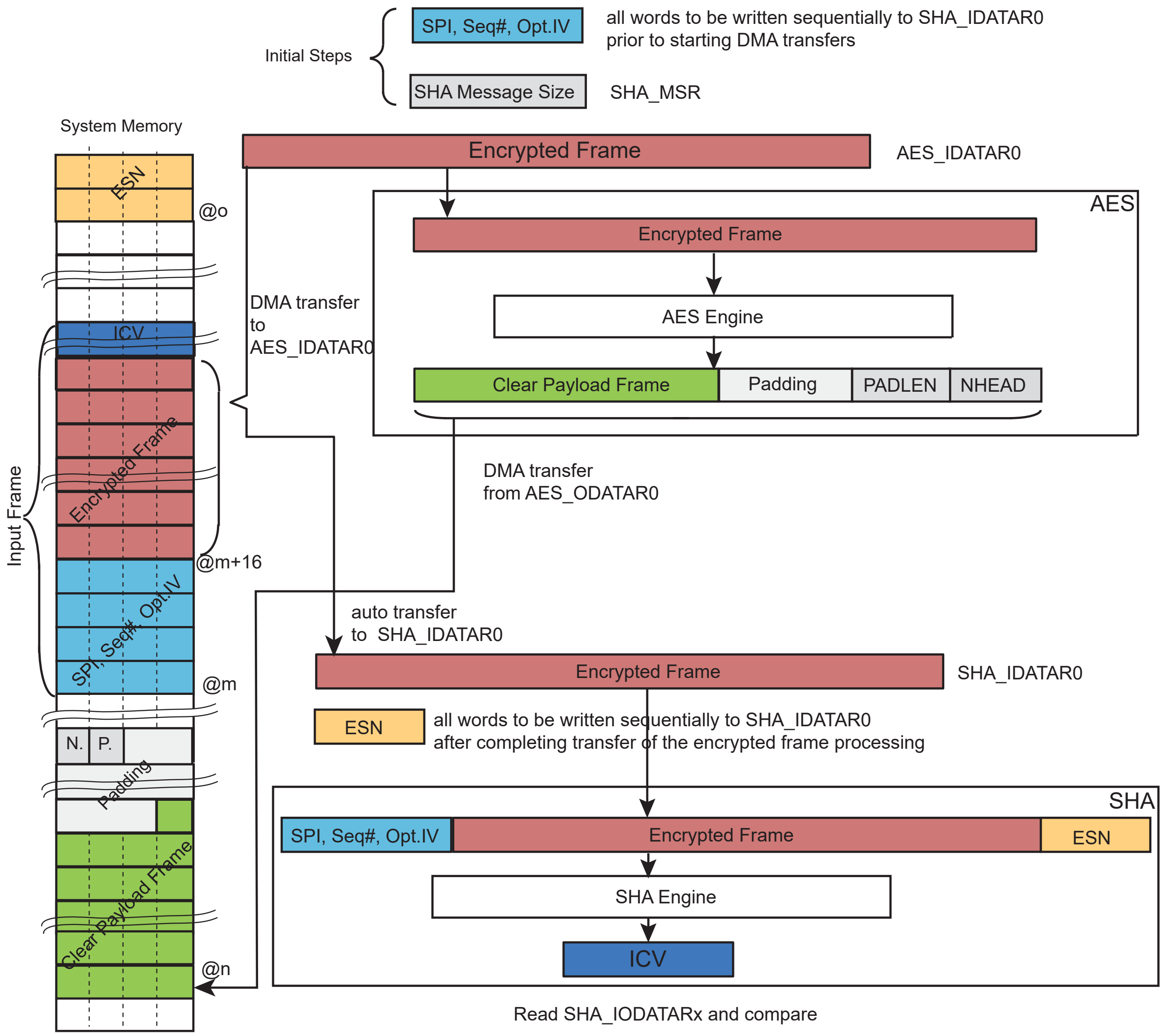
Figure 54-15. Decryption of an ESP IP Sec Frame without ESN



If the optional ESN trailer information is part of the ICV (see the following figure), the ESN must be manually written into SHA_IDATAR0. The ESN value must be written after completion of the system memory-to-AES DMA buffer transfer. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register.

When the optional ESN trailer information is part of the ICV, it is not possible to include the ICV received in the input frame to the first transfer descriptor. Moreover, if the HMAC algorithm is used for authentication, no automatic check can be performed when optimizing the processing performances of the SHA module. For more details, refer to the section “Secure Hash Algorithm (SHA)”. The result of the HMAC read in the SHA_IODATARx must be manually compared with the ICV value of the input frame. The comparison must be performed after the end of the authentication process. The authentication process is completed when the SHA_ISR.DATRDY flag is set.

Figure 54-16. Decryption of an ESP IPSec Frame with ESN



54.4.11 Security Features

54.4.11.1 Private Key Bus

The AES provides secure key transfer that requires a transfer command only, thus avoiding any manipulation of the key by software.

The AES features a set of Private Key internal registers that can be accessed only through the dedicated Private Key bus from the TRNG or OTPC.

The Private Key internal registers cannot be read from any peripheral or from software.

The AES key used by the encryption/decryption engine is either the Private Key internal registers content or the AES_KEYWRx registers loaded via the AES_KEYWRx.

To select the Private Key internal registers as the source of the AES key, AES_EMR.PKRS must be written to ‘1’.

When AES_EMR.PKRS is modified, it is mandatory to perform either a key write or a write in AES_CR.KSWP. The key write is mandatory when a new key value must be used. Writing AES_CR.KSWP to ‘1’ is mandatory if the key has been previously written and selected again after using another key.

If Private Key internal registers and software-loaded keys are already written, selecting one or the other requires only to configure AES_EMR.PKRS prior to writing AES_CR.KSWP=1.

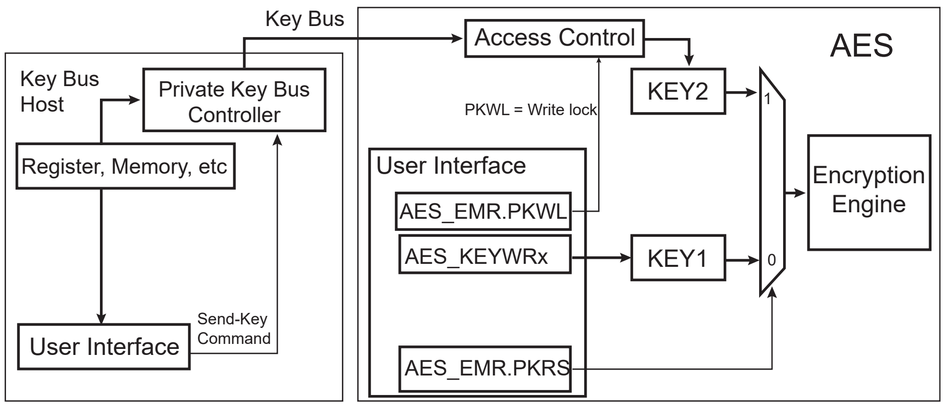
To write the Private Key internal registers, the software must:

1. Write a ‘1’ in AES_EMR.PKRS.
2. Trigger the key transfer over the Private Key bus from the TRNG or OTPC key bus host.
3. Wait for completion of the transfer signaled in the TRNG or OTPC Status register.
4. Check for any access violation in AES_WPSR.PKRPVS.

While AES_EMR.PKWL=0, it is possible to write the Private Key internal registers as many times as required.

As soon as AES_EMR.PKWL=1, the next write sequence on Private Key internal registers is the last one. Any additional write sequence in the Private Key internal registers has no effect, thus providing write-protection of these registers. A hardware reset is the only way to exit from the write-protected state.

Figure 54-17. Key Selection



54.4.11.2 Unspecified Register Access Detection

When an unspecified register access occurs, AES_ISR.URAD is raised. Its source is then reported in AES_ISR.URAT. Only the last unspecified register access is available through the AES_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- Input Data register written during the data processing when SMOD = IDATAR0_START
- Output Data register read during data processing
- Mode register written during data processing
- Output Data register read during sub-keys generation
- Mode register written during sub-keys generation
- Write-only register read access

AES_ISR.URAD and AES_ISR.URAT can only be reset by AES_CR.SWRST.

54.4.11.3 Clearing Key on Tamper Event

On a tamper detection event on WKUP1..8 pins, an immediate clear of the key (internal registers) can be performed if AES_MR.TAMPCLR=1. For configuration details, refer to the section “Real-Time Clock (RTC)”.

54.4.11.4 Register Write Protection

To prevent any single software error from corrupting AES behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the AES Write Protection Mode Register (AES_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the AES Write Protection Status Register (AES_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading AES_WPSR.

The following register(s) can be write-protected when WPEN is set in AES_WPMR:

- AES Mode Register
- AES Key Word Register x
- AES Initialization Vector Register x
- AES Additional Authenticated Data Length Register
- AES Plaintext/Ciphertext Length Register
- AES GCM Intermediate Hash Word Register x
- AES GCM H Word Register x
- AES Extended Mode Register
- AES Byte Counter Register
- AES Tweak Word Register x
- AES Alpha Word Register x

The following register(s) can be write-protected when WPITEN is set:

- AES Interrupt Enable Register
- AES Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- AES Control Register

54.4.11.5 Security and Safety Analysis and Reports

Several types of checks are performed when the AES is enabled.

The peripheral clock of the AES is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the AES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag AES_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the AES is also monitored and if an abnormal state is detected, the flag AES_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the AES are monitored and if an incorrect access is performed, the flag AES_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in AES_WPSR.SWETYP (see AES_WPSR for details). For example, writing the AES_ODATARx is an error, as well as reading the AES_IDATARx, when the AES_ISR.DATRDY flag is cleared. AES_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when AES_WPSR is read.

If one of these flags is set, the flag AES_ISR.SECE is set and can trigger an interrupt if the AES_IMR.SECE bit is ‘1’. SECE is cleared by reading AES_ISR.

It is possible to configure an action to be performed by AES as soon as an abnormal event detection occurs. If AES_WPMR.ACTION > 0, either a lock is performed or a lock and immediate clear of the AES_KEYWRx key. If a lock is performed, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see AES_MR.SMOD).

A locked state of the AES is unlocked as follows:

1. Read AES_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to AES_CR.UNLOCK.

It is possible to select the type of event that will lock the AES in case of abnormal event detection. See AES_WPMR.ACTION for details.

If the AES_MR.TMPCLR=1 and the tamper pin is active, the AES is locked.

54.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	AES_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0							KSWP	START
0x04	AES_MR	31:24	TAMPCLR							
		23:16	CKEY[3:0]					CFBS[2:0]		
		15:8	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
		7:0	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
0x08 ... 0x0F	Reserved									
0x10	AES_IER	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x14	AES_IDR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x18	AES_IMR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x1C	AES_ISR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8	URAT[3:0]							URAD
		7:0								DATRDY
0x20	AES_KEYWR0	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x24	AES_KEYWR1	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x28	AES_KEYWR2	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x2C	AES_KEYWR3	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x30	AES_KEYWR4	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x34	AES_KEYWR5	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x38	AES_KEYWR6	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	AES_KEYWR7	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x40	AES_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x44	AES_IDATAR1	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x48	AES_IDATAR2	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x4C	AES_IDATAR3	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x50	AES_ODATAR0	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54	AES_ODATAR1	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x58	AES_ODATAR2	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x5C	AES_ODATAR3	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x60	AES_IVR0	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x64	AES_IVR1	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x68	AES_IVR2	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x6C	AES_IVR3	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x70	AES_AADLENR	31:24	AADLEN[31:24]							
		23:16	AADLEN[23:16]							
		15:8	AADLEN[15:8]							
		7:0	AADLEN[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	AES_CLENR	31:24	CLEN[31:24]							
		23:16	CLEN[23:16]							
		15:8	CLEN[15:8]							
		7:0	CLEN[7:0]							
0x78	AES_GHASHR0	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x7C	AES_GHASHR1	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x80	AES_GHASHR2	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x84	AES_GHASHR3	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x88	AES_TAGR0	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x8C	AES_TAGR1	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x90	AES_TAGR2	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x94	AES_TAGR3	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x98	AES_CTRR	31:24	CTR[31:24]							
		23:16	CTR[23:16]							
		15:8	CTR[15:8]							
		7:0	CTR[7:0]							
0x9C	AES_GCMHR0	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA0	AES_GCMHR1	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA4	AES_GCMHR2	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA8	AES_GCMHR3	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xAC ... 0xAF	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB0	AES_EMR	31:24	BPE							
		23:16	NHEAD[7:0]							
		15:8	PADLEN[7:0]							
		7:0	PKRS	PKWL	PLIPD	PLIPEN			APM	APEN
0xB4	AES_BCNT	31:24	BCNT[31:24]							
		23:16	BCNT[23:16]							
		15:8	BCNT[15:8]							
		7:0	BCNT[7:0]							
0xB8 ... 0xBF	Reserved									
0xC0	AES_TWR0	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xC4	AES_TWR1	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xC8	AES_TWR2	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xCC	AES_TWR3	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xD0	AES_ALPHAR0	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xD4	AES_ALPHAR1	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xD8	AES_ALPHAR2	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xDC	AES_ALPHAR3	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xE0 ... 0xE3	Reserved									
0xE4	AES_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	AES_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS

54.5.1 AES Control Register

Name: AES_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
							KSWP	START
Access							W	W
Reset							–	–

Bit 24 – UNLOCK Unlock Processing
AES_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if AES_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the AES. A software-triggered reset of the AES interface is performed.

Bit 1 – KSWP Key Swap

Value	Description
0	No effect.
1	Activates the set of key registers defined by AES_EMR if AES_EMR.PKRS has been changed.

Bit 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual encryption/decryption process.

54.5.2 AES Mode Register

Name: AES_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
	CKEY[3:0]					CFBS[2:0]		
Access	W	W	W	W		R/W	R/W	R/W
Reset	0	0	0	–		0	0	0

Bit	15	14	13	12	11	10	9	8
	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 31 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on the AES_KEYWRx key.
1	A tamper detection event immediately clears the AES_KEYWRx key.

Bits 23:20 – CKEY[3:0] Key

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time AES_MR is programmed. For subsequent programming of AES_MR, any value can be written, including that of 0xE. Always reads as 0.

Bits 18:16 – CFBS[2:0] Cipher Feedback Data Size

Value	Name	Description
0	SIZE_128BIT	128-bit
1	SIZE_64BIT	64-bit
2	SIZE_32BIT	32-bit
3	SIZE_16BIT	16-bit
4	SIZE_8BIT	8-bit

Bit 15 – LOD Last Output Data Mode

⚠WARNING In DMA mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable results.

Value	Description
0	No effect. After each end of encryption/decryption, the output data are available either on the output data registers (Manual and Auto modes) or at the address specified in the Channel Buffer Transfer Descriptor for DMA mode. In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.
1	The DATRDY flag is cleared when at least one of the Input Data Registers is written. No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

Bits 14:12 – OPMOD[2:0] Operating Mode
For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.
When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

Value	Name	Description
0	ECB	ECB: Electronic Codebook mode
1	CBC	CBC: Cipher Block Chaining mode
2	OFB	OFB: Output Feedback mode
3	CFB	CFB: Cipher Feedback mode
4	CTR	CTR: Counter mode (16-bit internal counter)
5	GCM	GCM: Galois/Counter mode
6	XTS	XTS: XEX-based tweaked-codebook mode

Bits 11:10 – KEYSIZE[1:0] Key Size

Value	Name	Description
0	AES128	AES Key Size is 128 bits
1	AES192	AES Key Size is 192 bits
2	AES256	AES Key Size is 256 bits

Bits 9:8 – SMOD[1:0] Start Mode
If a DMA transfer is used, configure SMOD to 2. See DMA Mode for more details.

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	AES_IDATAR0 access only Auto Mode (DMA)

Bits 7:4 – PROCDLY[3:0] Processing Delay
Processing Time = N × (PROCDLY + 1)
where

- N = 10 when KEYSIZE = 0
- N = 12 when KEYSIZE = 1
- N = 14 when KEYSIZE = 2

The processing time represents the number of clock cycles that the AES needs in order to perform one encryption/ decryption.
Note: The best performance is achieved with PROCDLY equal to 0.

Bit 3 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	AES_IDATARx cannot be written during processing of previous block.
1	ACTIVE	AES_IDATARx can be written during processing of previous block when SMOD = 2. It speeds up the overall runtime of large files.

Bit 1 – GTAGEN GCM Automatic Tag Generation Enable

Value	Description
0	Automatic GCM Tag generation disabled.
1	Automatic GCM Tag generation enabled.

Bit 0 – CIPHER Processing Mode

Value	Description
0	Decrypts data.
1	Encrypts data.

54.5.3 AES Interrupt Enable Register

Name: AES_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event Interrupt Enable

Bit 18 – PLENERR Padding Length Error Interrupt Enable

Bit 17 – EOPAD End of Padding Interrupt Enable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

54.5.4 AES Interrupt Disable Register

Name: AES_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event Interrupt Disable

Bit 18 – PLENERR Padding Length Error Interrupt Disable

Bit 17 – EOPAD End of Padding Interrupt Disable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

54.5.5 AES Interrupt Mask Register

Name: AES_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event Interrupt Mask

Bit 18 – PLENERR Padding Length Error Interrupt Mask

Bit 17 – EOPAD End of Padding Interrupt Mask

Bit 16 – TAGRDY GCM Tag Ready Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

54.5.6 AES Interrupt Status Register

Name: AES_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	URAT[3:0]							URAD
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	There is no security report in AES_WPSR.
1	One security flag is set in AES_WPSR.

Bit 18 – PLENERR Padding Length Error

Value	Description
0	No Padding Length Error occurred.
1	Padding Length Error detected.

Bit 17 – EOPAD End of Padding

Value	Description
0	Padding is not over.
1	Padding phase is over.

Bit 16 – TAGRDY GCM Tag Ready

Value	Description
0	GCM Tag is not valid.
1	GCM Tag generation is complete (cleared by reading GCM Tag, starting another processing or when writing a new key).

Bits 15:12 – URAT[3:0] Unspecified Register Access (cleared by writing SWRST in AES_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name	Description
0	IDR_WR_PROCESSING	Input Data register written during the data processing when SMOD = 2 mode.
1	ODR_RD_PROCESSING	Output Data register read during the data processing.
2	MR_WR_PROCESSING	Mode register written during the data processing.
3	ODR_RD_SUBKGEN	Output Data register read during the sub-keys generation.

Value	Name	Description
4	MR_WR_SUBKGEN	Mode register written during the sub-keys generation.
5	WOR_RD_ACCESS	Write-only register read access.

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing SWRST in AES_CR)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 0 – DATRDY Data Ready (cleared by setting bit START or bit SWRST in AES_CR or by reading AES_ODATARx)

Value	Description
0	Output data not valid.
1	Encryption or decryption process is completed.

Note: If AES_MR.LOD = 1: In Manual and Auto mode, the DATRDY flag can also be cleared by writing at least one AES_IDATARx.

54.5.7 AES Key Word Register x

Name: AES_KEYWRx
Offset: 0x20 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

These registers are write-only to prevent the key from being read by another application.

Note: AES_KEYWRx registers are not used if the Private Key internal registers are selected (AES_EMR.PKRS=1).

Bit	31	30	29	28	27	26	25	24
	KEYW[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	KEYW[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	KEYW[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	KEYW[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEYW[31:0] Key Word
The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for AES encryption/decryption.
AES_KEYWR0 corresponds to the first word of the key and respectively AES_KEYWR3/AES_KEYWR5/AES_KEYWR7 to the last one.
Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
These registers are write-only to prevent the key from being read by another application.
Note: To write AES_KEYWRx and start using the key immediately, AES_EMR.PKRS must be written to 0 prior to writing AES_KEYWRx.

54.5.8 AES Input Data Register x

Name: AES_IDATARx
Offset: 0x40 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data Word
The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.
AES_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, and AES_IDATAR3 to the last one.
These registers are write-only to prevent the input data from being read by another application.

54.5.9 AES Output Data Register x

Name: AES_ODATARx
Offset: 0x50 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted. AES_ODATAR0 corresponds to the first word, AES_ODATAR3 to the last one.

54.5.10 AES Initialization Vector Register x

Name: AES_IVRx
Offset: 0x60 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.

AES_IVR0 corresponds to the first word of the Initialization Vector, AES_IVR3 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

For CBC, OFB and CFB modes, the IV input value corresponds to the initialization vector.

For CTR mode, the IV input value corresponds to the initial counter value.

These registers are not used in ECB mode and must not be written.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, AES_IVRx must be cleared before switching to the new mode

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector

54.5.11 AES Additional Authenticated Data Length Register

Name: AES_AADLENR
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	AADLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AADLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AADLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AADLEN[31:0] Additional Authenticated Data Length
Length in bytes of the Additional Authenticated Data (AAD) that is to be processed.
Note: The maximum byte length of the AAD portion of a message is limited to the 32-bit counter length.

54.5.12 AES Plaintext/Ciphertext Length Register

Name: AES_CLENR
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CLEN[31:0] Plaintext/Ciphertext Length
Length in bytes of the plaintext/ciphertext (C) data that is to be processed.
Note: The maximum byte length of the C portion of a message is limited to the 32-bit counter length.

54.5.13 AES GCM Intermediate Hash Word Register x

Name: AES_GHASHRx
Offset: 0x78 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GHASH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GHASH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GHASH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GHASH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GHASH[31:0] Intermediate GCM Hash Word x
The four 32-bit Intermediate Hash Word registers expose the intermediate GHASH value. May be read to save the current GHASH value so processing can later be resumed, presumably on a later message fragment. Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
If an application software-specific hash initial value is needed for the GHASH, it must be written to AES_GHASHRx:

- after writing AES_KEYWRx, if any
- before starting the input data feed.

54.5.14 AES GCM Authentication Tag Word Register x

Name: AES_TAGRx
Offset: 0x88 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TAG[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAG[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TAG[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TAG[31:0] GCM Authentication Tag x
The four 32-bit Tag registers contain the final 128-bit GCM Authentication tag (*T*) when GCM processing is complete. TAG0 corresponds to the first word, TAG3 to the last word.

54.5.15 AES GCM Encryption Counter Value Register

Name: AES_CTRR
Offset: 0x98
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CTR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CTR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CTR[31:0] GCM Encryption Counter
Reports the current value of the 32-bit GCM counter.

54.5.16 AES GCM H Word Register x

Name: AES_GCMHRx
Offset: 0x9C + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	H[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	H[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	H[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	H[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – H[31:0] GCM H Word x
The four 32-bit H Word registers contain the 128-bit GCM hash subkey *H*value.
Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey *H* generation
- AES_GHASHRx Clear

If the application software requires a specific hash subkey, the automatically-generated H value can be overwritten in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation for details.
Generating a GCM hash subkey *H* by a write in AES_GCMHRx enables to:

- select the GCM hash subkey *H* for GHASH operations,
- select one operand to process a single GF128 multiply.

54.5.17 AES Extended Mode Register

Name: AES_EMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BPE							
Access	R/W							
Reset	0							
Bit	23	22	21	20	19	18	17	16
	NHEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PKRS	PKWL	PLIPD	PLIPEN			APM	APEN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit 31 – BPE Block Processing End

Value	Description
0	AES_ISR.DATRDY flag reports only the end message encryption processing. No intermediate block processing is reported when SMOD=2. When a DMA is used to transfer data, BPE must be cleared.
1	AES_ISR.DATRDY flag reports each end of block processing when SMOD=2. When AES_IDATARx are not loaded by a DMA and SMOD=2, this bit can be written to 1 to rise the AES_ISR.DATRDY flag when a new data block can be written.

Bits 23:16 – NHEAD[7:0] IPSec Next Header

Value	Description
0-255	IPSec Next Header field

Bits 15:8 – PADLEN[7:0] Auto Padding Length

Value	Description
0-255	Padding length in bytes

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The key used by the AES is in the AES_KEYWRx registers.
1	The key used by the AES is in the Private Key internal registers written through the Private Key bus.

Bit 6 – PKWL Private Key Write Lock

Once PKWL is set to ‘1’, only a hardware reset sets this bit to ‘0’ internally. Writing it to '0' with a register access has no impact (although the field will be read to value ‘0’).

Value	Description
0	The Private Key internal registers can be written multiple times via the Private Key bus.
1	The Private Key internal registers can be written only once via the Private Key bus until hardware reset.

Bit 5 – PLIPD Protocol Layer Improved Performance Decipher

Value	Description
0	Protocol layer improved performance is in ciphering mode.
1	Protocol layer improved performance is in deciphering mode.

Bit 4 – PLIPEN Protocol Layer Improved Performance Enable

Value	Description
0	Protocol layer improved performance is disabled.
1	Protocol layer improved performance is enabled.

Bit 1 – APM Auto Padding Mode

Value	Description
0	Auto Padding performed according to IPSec standard.
1	Auto Padding performed according to SSL standard.

Bit 0 – APEN Auto Padding Enable

Value	Description
0	Auto Padding feature is disabled.
1	Auto Padding feature is enabled.

54.5.18 AES Byte Counter Register

Name: AES_BCNT
Offset: 0xB4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BCNT[31:0] Auto Padding Byte Counter
Auto padding byte counter value. BCNT must be greater than 0.

54.5.19 AES Tweak Word Register x

Name: AES_TWRx
Offset: 0xC0 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TWEAK[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TWEAK[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TWEAK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TWEAK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TWEAK[31:0] Tweak Word x
The four 32-bit Tweak Word registers contain the 128-bit Tweak value.

54.5.20 AES Alpha Word Register x

Name: AES_ALPHARx
Offset: 0xD0 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ALPHA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	ALPHA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	ALPHA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ALPHA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – ALPHA[31:0] Alpha Word x
The four 32-bit Alpha Word registers contain the 128-bit primitive of GF(2^{128}) to use for the first processing.

54.5.21 AES Write Protection Mode Register

Name: AES_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414553	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection
When the field AES_WPMR.ACTION differs from 0 and an abnormal event or internal state is detected, the AES is locked until the unlock command is issued (AES_CR.UNLOCK=1). The lock source must be cleared before performing the unlock command. If AES_WPSR.SEQE=1, the following two actions must be performed:
1/ Read AES_WPSR.
2/ Issue software reset by writing a 1 in AES_CR.SWRST.
A specific configuration applies where the sequence does not clear the lock source (AES_WPSR=0).
If AES_WPSR.SEQE remains high after the clearing sequence, then only a hardware reset will unlock the AES. A hardware reset can be performed by issuing a reset controller software reset (refer to the section "Reset Controller (RSTC)"). This condition can be met when AES_EMR.PKWL=1 and a key has been loaded through the Private Key bus. The key loaded through the key bus is corrupted, but it is impossible to reload a new key unless a hardware reset is issued.

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags is set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.

Value	Name	Description
3	LOCK_ANY_EV	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
5	CLEAR_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in AES_WPSR.WPVSRC and the last software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in AES_WPSR.WPVSRC and only the first software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

54.5.22 AES Write Protection Status Register

Name: AES_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the AES is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	AES is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of AES_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A key write, init value write, output data read, AES_MR and AES_EMR write, GCM configuration registers write, AES_TWRx and AES_ALPHARx registers write, AES_BCNT write, Private Key Bus access has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the key is not fully loaded into the AES_KEYWRx registers.

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Internal Register Protection Violation Status (cleared on read)

Value	Description
0	No Private Key Internal Register access violation has occurred since the last read of AES_WPSR.
1	A Private Key Internal Register access violation has occurred since the last read of AES_WPSR.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of AES_WPSR.
1	A software error has occurred since the last read of AES_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of AES_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of AES_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of AES_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of AES_WPSR. This flag can only be set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of AES_WPSR.
1	A write protect violation has occurred since the last read of AES_WPSR. The address offset of the violated register is reported into field WPVSRC.

55. Secure Hash Algorithm (SHA)

55.1 Description

The Secure Hash Algorithm (SHA) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-4* specification.

The 512/1024-bit block of message is respectively stored in 16/32 x 32-bit registers, (SHA_IDATARx/ SHA_IODATARx) which are write-only.

As soon as the input data is written, hash processing can be started. The registers comprising the block of a message must be entered consecutively. Then, after the processing period, the message digest is ready to be read out on the 5 up to 8/16 x 32-bit output data registers (SHA_IODATARx) or through the DMA channels.

The SHA supports the SHA512 derivative algorithms SHA512/224 and SHA512/256 which are based on the SHA512 algorithm with specific initial vectors and a truncated digest. Unless specified otherwise, features and functionality of the SHA512 algorithm also apply to the SHA512/224 and SHA512/256 algorithms.

The SHA supports the HMAC-SHA512 derivative algorithms HMAC-SHA512/224 and HMAC-SHA512/256 which are based on the HMAC-SHA512 algorithm with specific initial vectors and a truncated digest. Unless specified otherwise, features and functionality of the HMAC-SHA512 algorithm also apply to the HMAC-SHA512/224 and HMAC-SHA512/256 algorithms.

55.2 Embedded Characteristics

- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256, SHA384, SHA512, SHA512/224, SHA512/256)
- Supports Hash-based Message Authentication Code (HMAC) Algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, HMAC-SHA512/224, HMAC-SHA512/256)
- Compliant with FIPS Publication 180-4
- Supports Automatic Padding of Messages
- Supports Up to 2 Sets of Initial Hash Values Registers (HMAC Acceleration or other)
- Supports Automatic Check of the Hash (HMAC Acceleration or other)
- Tightly Coupled to AES for Protocol Layers Improved Performances
- Configurable Processing Period:
  - 85 clock cycles to obtain a fast SHA1 runtime, 88 clock cycles for SHA384, SHA512 or 209 Clock Cycles for Maximizing Bandwidth of Other Applications
  - 72 clock cycles to obtain a fast SHA224, SHA256 runtime or 194 clock cycles for maximizing bandwidth of other applications
- Connection to DMA Channel Capabilities Optimizes Data Transfers
- Double Input Buffer Optimizes Runtime
- Register Write Protection

55.3 Product Dependencies

55.3.1 Power Management

The SHA may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SHA clock.

55.3.2 Interrupt Sources

The SHA interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by the Interrupt Mask register (SHA_IMR) and the Interrupt Status register (SHA_ISR), whereas the interrupt line for safety/security functions is driven by Write Protection Status register (SHA_WPSR) flags. If one of the flags SHA_WPSR.WPVS, SHA_WPSR.CGD, SHA_WPSR.SEQE or SHA_WPSR.SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the SHA.

55.4 Functional Description

The Secure Hash Algorithm (SHA) module requires a padded message according to the FIPS 180 specification. This message can be provided with the padding to the SHA module, or the padding can be automatically computed by the SHA module if the size of the message is provided. The first block of the message must be indicated to the module by a specific command. The SHA module produces an N-bit message digest each time a block is written and processing period ends, where N is 160 for SHA1, 224 for SHA224, 256 for SHA256, 384 for SHA384, 512 for SHA512. The SHA module is also capable of computing a Hash-based Message Authentication Code (HMAC) algorithm.

55.4.1 SHA Algorithm

The SHA can process SHA1, SHA224, SHA256, SHA384, SHA512 by configuring the ALGO field in the Mode register (SHA_MR).

55.4.2 HMAC Algorithm

The HMAC algorithm is as follows:

$HMAC_K(m) = h((K_0 \oplus opad) \| h((K_0 \oplus ipad) \| m))$

where:

- h = SHA function
- K_0 = the key K after any necessary pre-processing to form a block size key
- m = message to authenticate
- || = concatenation operator
- $\oplus$ = XOR operator
- ipad = predefined constant (0x3636...3636)
- opad = predefined constant (0x5C5C...5C5C)

The SHA provides a fully optimized processing of the HMAC algorithm by executing the following operations:

- starting the SHA algorithm from any user predefined hash value, thus ‘$h(K_0 \oplus ipad)$’ for first HMAC hash and ‘$h(K_0 \oplus opad)$’ for second HMAC hash
- performing automatic padding
- routing automatically the first hash result ‘$h((K_0 \oplus ipad) \| m)$’ to the source of the second hash processing ‘$h((K_0 \oplus opad) \| (\text{first hash result}))$’ including the concatenation of the first hash result to ‘$K_0 \oplus opad$’.

To perform the HMAC operation, the ALGO field value must be greater than 7, the automatic padding feature must be enabled (MSGSIZE and BYTCNT fields differ from 0) and the SHA internal initial hash value registers 0 and 1 must be configured, respectively, with the hash results of input blocks “$K_0 \oplus ipad$” and “$K_0 \oplus opad$” (see Internal Registers for Initial Hash Value or Expected Hash Result).

The size of the message (‘m’) must be written in the MSGSIZE and BYTCNT fields.

The FIRST bit in the SHA Control register (SHA_CR) should be set before writing the first block of the message.

The SHA can process HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 by configuring the SHA_MR.ALGO field.

55.4.3 Processing Period

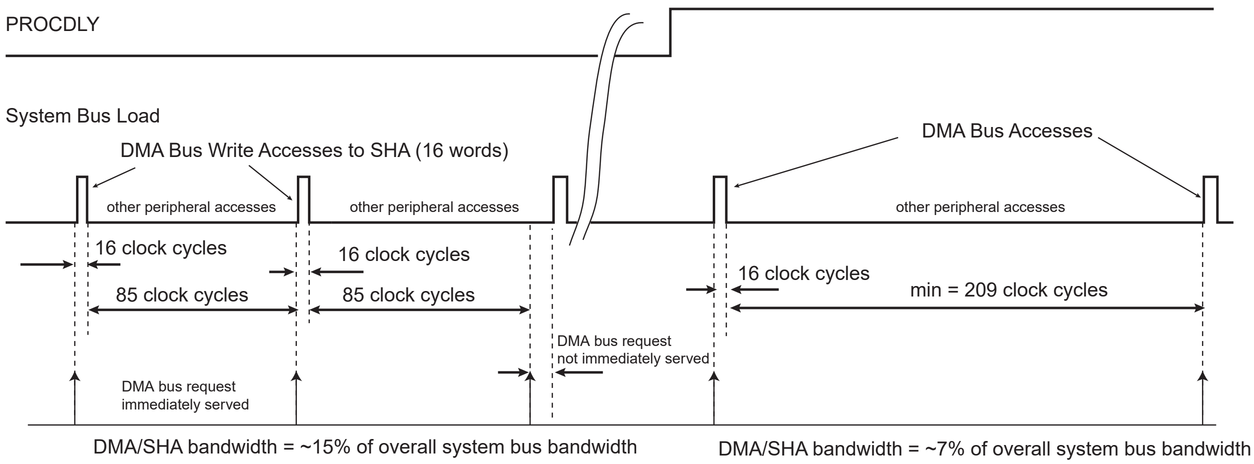
When SHA is enabled and DMA is used to write the messages, the inherent processing period may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, other DMA channels). The SHA engine inherent processing period can be configured to reduce the bandwidth required by writing SHA_MR.PROCDLY=1.

In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 209 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1 (see the figure below).

In SHA256 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 194 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1.

In SHA384 or SHA512 mode, the shortest processing period is 88 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

Figure 55-1. Bandwidth Usage in SHA-1 Mode



55.4.4 Double Input Buffer

The SHA Input Data registers (SHA_IDATARx) can be double-buffered to reduce the runtime of large messages.

Double-buffering allows a new message block to be written while the previous message block is being processed. This is only possible when DMA accesses are performed (SMOD = 2).

The DUALBUFF bit in the SHA_MR must be set to have double input buffer access.

55.4.5 Internal Registers for Initial Hash Value or Expected Hash Result

The SHA module embeds two sets of internal registers (IR0, IR1) to store different data used by the SHA or HMAC algorithms (see the figure User Initial Hash Value and Expected Hash Internal Register Access). These internal registers are accessed through SHA Input Data registers (SHA_IDATARx).

When the ALGO field selects SHA algorithms, IR0 can be configured with a user initial hash value. This initial hash value can be used to compute a custom hash algorithm with two sets of different initial constants, or to continue a hash computation by providing the intermediate hash value previously returned by the SHA module.

When the ALGO field selects SHA algorithms, IR1 can be configured with either a user initial hash value or an expected hash result. The expected hash result must be configured in the IR1 if the field CHECK = 1 (see Automatic Check). If the field CHECK = 0 or 2, IR1 can be configured with a user initial hash value that differs from IR0 value.

When the ALGO field selects HMAC algorithms, IR0 must be configured with the hash result of $K_0 \oplus$ ipad and IR1 must be configured with the hash result of $K_0 \oplus$ opad. These pre-computed first blocks speed up the HMAC computation by saving the time to compute the intermediate hash values of the first block which is constant while the secret key is constant (see HMAC Algorithm).

Table 55-1. Configuration Values of Internal Registers

Register	SHA Modes (ALGO < 8)			HMAC Modes (ALGO > 7)
	CHECK = 0	CHECK = 1	CHECK = 2	
IR0	User Initial Hash	User Initial Hash	User Initial Hash	hash($K_0 \oplus$ ipad)
IR1	User Initial Hash	Expected Hash Result	User Initial Hash	hash($K_0 \oplus$ opad)

To calculate the initial HMAC values, follow this sequence:

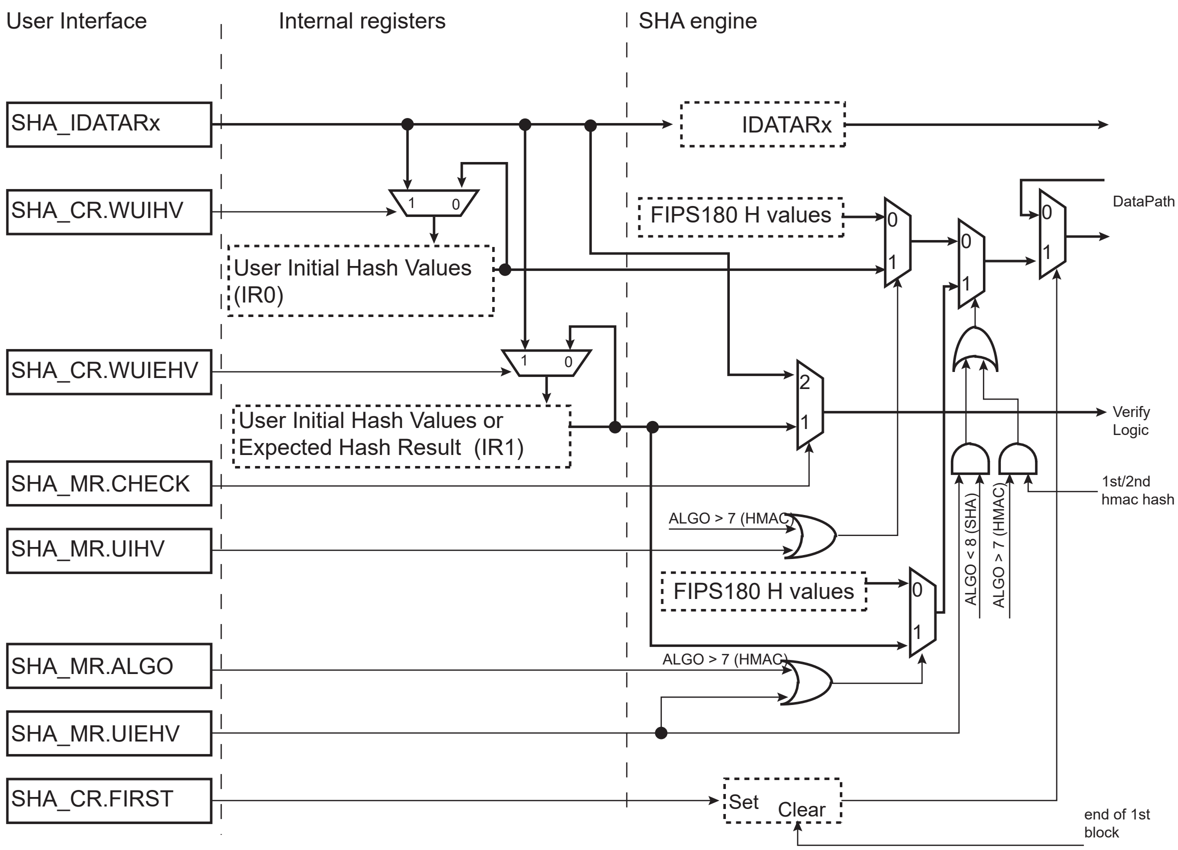
1. Calculate K_0.
2. Calculate $K_0 \oplus$ ipad and $K_0 \oplus$ opad.
3. Perform a hash of the result of $K_0 \oplus$ ipad and $K_0 \oplus$ opad (auto-padding must be disabled for that type of hash).
4. Write h($K_0 \oplus$ ipad) and h($K_0 \oplus$ opad) in IR0 and IR1 respectively.

To write IR0 or IR1, follow this sequence:

1. Set SHA_CR. WUIHV (IR0) or SHA_CR.WUIEHV (IR1).
2. Write the data in SHA_IDATARx. The number of registers to write depends on the type of data (user initial hash values or expected hash result) and on the type of algorithm selected:
   – For user initial hash values:
     • SHA_IDATAR0 to SHA_IDATAR4 for SHA1
     • SHA_IDATAR0 to SHA_IDATAR7 for SHA224 or SHA256
     • SHA_IDATAR0 to SHA_IDATAR15 for SHA384, SHA512, SHA512/224, SHA512/256
   – For expected hash result:
     • SHA_IDATAR0 to SHA_IDATAR4 for SHA1
     • SHA_IDATAR0 to SHA_IDATAR6 for SHA224 or SHA512/224
     • SHA_IDATAR0 to SHA_IDATAR7 for SHA256 or SHA512/256
     • SHA_IDATAR0 to SHA_IDATAR11 for SHA384
     • SHA_IDATAR0 to SHA_IDATAR16 for SHA512
3. Clear SHA_CR.WUIHV or SHA_CR.WUIEHV.

IR0 and IR1 are automatically selected for HMAC processing if the field ALGO selects HMAC algorithms. If SHA algorithms are selected, the internal registers are selected if the corresponding UIHV or UIEHV bits are set.

Figure 55-2. User Initial Hash Value and Expected Hash Internal Register Access



55.4.6 Automatic Padding

The SHA module features an automatic padding computation to speed up the execution of the algorithm.

The automatic padding function requires the following information:

- Complete message size in bytes to be written in the MSGSIZE field of the SHA Message Size register (SHA_MSR).
  The size of the message is written at the end of the last block, as required by the FIPS 180 specification (the size is automatically converted into a bit-size).
  Note: SHA_MSR is a 32-bit register, thus the automatic padding capability is limited to messages of less than 4 gigabytes. For messages greater than 4 gigabytes, padding must be performed by the software.
- Number of remaining bytes (to write in the SHA_IDATARx) to be written in the BYTCNT field of the SHA Bytes Count register (SHA_BCR).
  Automatic padding occurs when the BYTCNT field reaches 0. At each write in the SHA Input registers, the BYTCNT field value is decreased by the number of bytes written.

The BYTCNT field value must be written with the same value as the MSGSIZE field value if the full message is processed. If the message is partially preprocessed and an initial hash value is used, BYTCNT must be written with the remaining bytes to hash while MSGSIZE holds the message size.

To disable the automatic padding feature, the MSGSIZE and BYTCNT fields must be configured with 0.

55.4.7 Automatic Check

The SHA module features an automatic check of the hash result with the expected hash. A check failure can generate an interrupt if configured in the SHA Interrupt Enable register (SHA_IER).

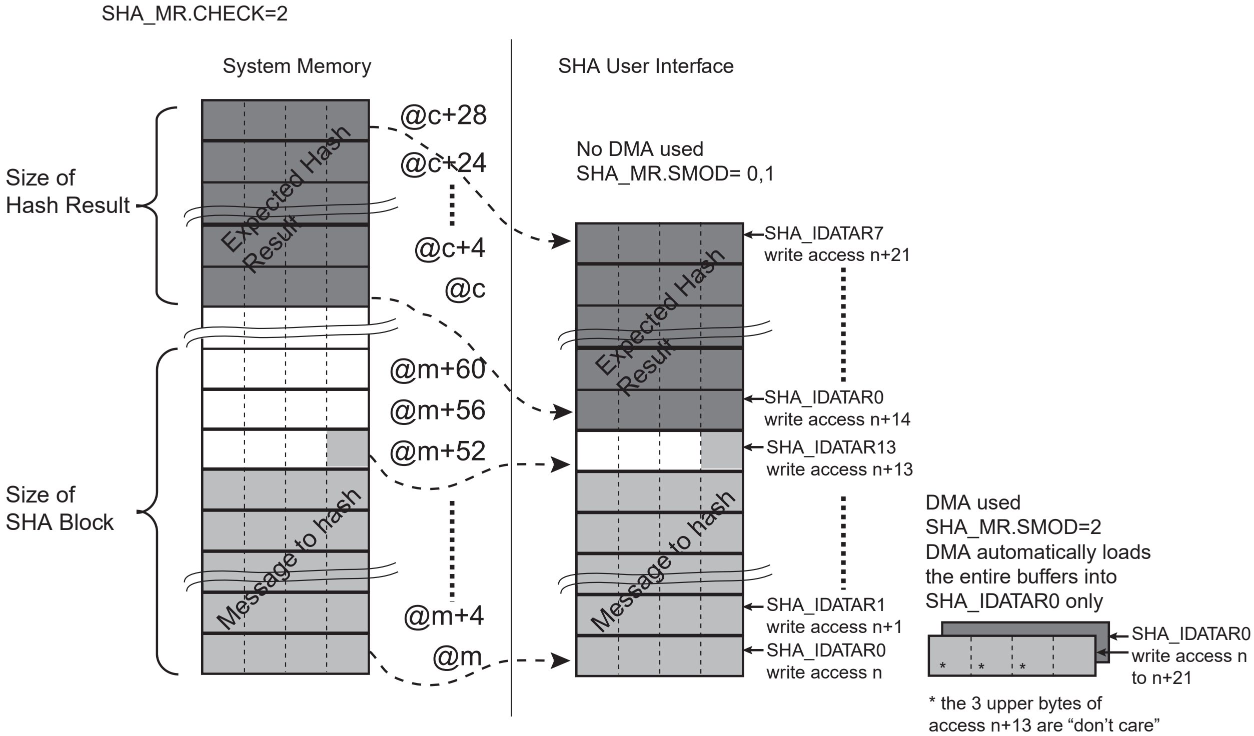
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

There are two methods to configure the expected hash result:

- if SHA_MR.CHECK = 1, the expected hash result is read from the internal register (IR1). This method cannot be used when HMAC algorithms is selected because this register is already used to store user initial hash values for the second hash processing. IR1 cannot be read by software.
- If SHA_MR.CHECK = 2, the expected hash result is written in the SHA_IDATARx after the message.

When SHA_MR.CHECK = 2, the method can provide more flexibility of use if a message is stored in system memory together with its expected hash result. A DMA with linked list can be used to ease the transfer of the message and its expected hash result.

Figure 55-3. Message and Expected Hash Result Memory Mapping



The number of 32-bit words of the hash result to check with the expected hash can be selected with SHA_MR.CHKCNT. The status of the check is available in the CHKST field in the SHA Interrupt Status register (SHA_ISR).

An interrupt can be generated (if enabled) when the check is completed. The check occurs several clock cycles after the computation of the requested hash, so the interrupt and the CHECKF bit are set several clock cycles after the DATRDY flag of the SHA_ISR.

55.4.8 Protocol Layers Improved Performances

The SHA can be tightly coupled to the AES module to improve performances when processing protocol layers such as IPsec or OpenSSL.

When the AES is configured to be tightly coupled to SHA (AES_MR), SHA must be always configured in Double Buffer mode (SHA_MR.DUALBUFF = 1).

Refer to the section "Advanced Encryption Standard (AES)" for details.

55.4.9 Start Modes

SHA_MR.SMOD is used to select the Hash Processing Start mode.

55.4.9.1 Manual Mode

In Manual mode, the sequence is as follows:

1. Set SHA_IER.DATRDY (Data Ready) , depending on whether an interrupt is required at the end of processing.
2. If the initial hash values differ from the FIPS standard, set SHA_MR.UIHV and/or SHA_MR.UIEHV. If the initial hash values comply with the FIPS180 specification, clear SHA_MR.UIHV and/or SHA_MR.UIEHV.
3. If automatic padding is required, configure SHA_MSR.MSGSIZE with the number of bytes of the message, and configure SHA_BCR.BYTCNT with the remaining number of bytes to write. The BYTCNT field must be written with a value different from MSGSIZE field value if the message is preprocessed and completed by using user initial hash values.
   If automatic padding is not required, configure SHA_MSR.MSGSIZE and SHA_BCR.BYTCNT to 0.
4. The FIRST command must be set by writing a 1 into the corresponding bit of the Control register (SHA_CR) to start a hash computation with initial constants (first block of a message) or to resume after message

processing was interrupted. When a first message processing is interrupted to process another message, the intermediate hash results must be stored in the system memory and they must be reloaded in user initial values registers (IR0 accessed via SHA_IDATAR when SHA_CR.WUIHV=1) prior to resume and continue the processing of the first message. For the other blocks, there is nothing to write.

5. Write the block to process in SHA_IDATARx.
6. To begin processing, set SHA_CR.START.
7. When processing is completed, the bit DATRDY in the Interrupt Status register (SHA_ISR) rises. If an interrupt has been enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated.
8. Repeat the write procedure for each block (step 5), start procedure (step 6) and wait for the interrupt procedure (step 7) up to the last block of the entire message. Each time the start procedure is complete, the DATRDY flag is cleared.
9. After the last block is processed (the DATRDY flag is set, if an interrupt was enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated), read the message digest in the Output Data registers. The DATRDY flag is automatically cleared when reading the SHA_IODATARx registers.

55.4.9.2 Auto Mode

In Auto mode, processing starts as soon as the correct number of SHA_IDATARx is written. No action is required in SHA_CR.

55.4.9.3 DMA Mode

The DMA can be used in association with the SHA to perform the algorithm on a complete message without any action by the software during processing.

SHA_MR.SMOD must be configured to 2.

The DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be set to point to the SHA_IDATAR0.

The DMA chunk size must be set to transfer, for each trigger request, 16 words of 32 bits.

The FIRST bit of SHA_CR must be set before starting the DMA when the first block is transferred.

Note: The FIRST bit command is also used to resume after message processing was interrupted. When a first message processing is interrupted to process another message, the intermediate hash results must be stored in the system memory and they must be reloaded in user initial values registers (IR0 accessed via SHA_IDATAR when SHA_CR.WUIHV=1) prior to resume and continue the processing of first message.. Thus, the DMA data buffers and SHA_CR.FIRST command must be managed accordingly.

The DMA generates an interrupt when the end of buffer transfer is completed but the SHA processing is still in progress. The end of SHA processing is indicated by the flag DATRDY in the SHA_ISR.

If automatic padding is disabled, the end of SHA processing requires two interrupts to be verified. The DMA end of transfer interrupt must be verified first, then the SHA DATRDY interrupt must be enabled and verified (see the figure Interrupts Processing with DMA).

If automatic padding is enabled, the end of SHA processing requires only one interrupt to be verified. The DMA end of transfer is not required, so the SHA DATRDY interrupt must be enabled prior to start the DMA and DATRDY interrupt is the only one to be verified (see the figure Interrupts Processing with DMA and Automatic Padding).

Figure 55-4. Interrupts Processing with DMA

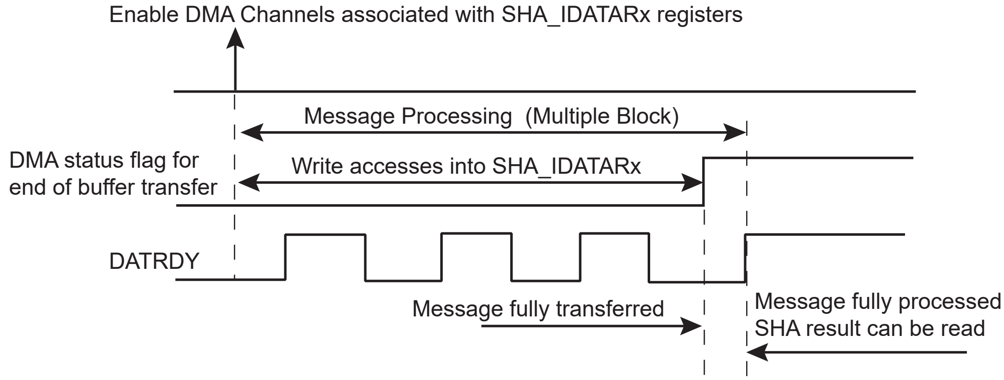
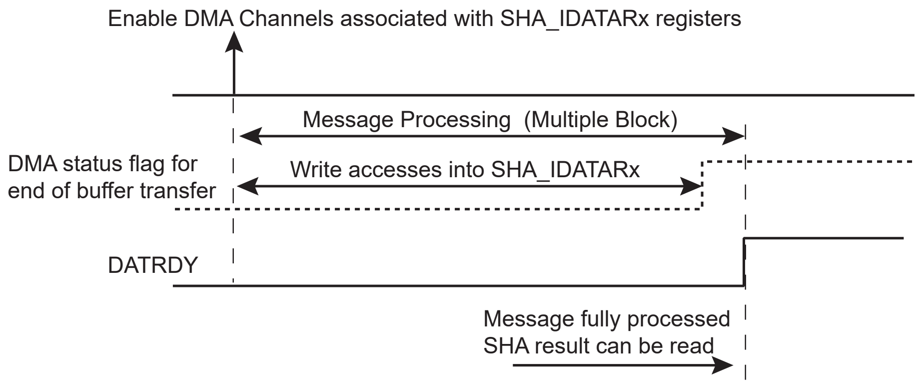


Figure 55-5. Interrupts Processing with DMA and Automatic Padding



55.4.9.4 SHA Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The SHA interface requires little-endian format words. However, in accordance with the protocol of the FIPS 180 specification, data is collected, processed and stored by the SHA algorithm in big-endian form.

The following example illustrates how to configure the SHA:

If the first 64 bits of a message (according to FIPS 180, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then the SHA_IDATAR0 and SHA_IDATAR1 registers must be written with the following pattern:

- SHA_IDATAR0 = 0xcadefeca
- SHA_IDATAR1 = 0x67452301

In a little-endian system, the message (according to FIPS 180) starting with pattern 0xcafedeca_01234567 is stored into memory as follows:

- 0xca stored at initial offset (for example 0x00),
- then 0xfe stored at initial offset + 1 (i.e., 0x01),
- 0xde stored at initial offset + 2 (i.e., 0x02),
- 0xca stored at initial offset + 3 (i.e., 0x03).

If the message is received through a serial-to-parallel communication channel, the first received character is 0xca and it is stored at the first memory location (initial offset). The second byte, 0xfe, is stored at initial offset + 1.

When reading on a 32-bit little-endian system bus, the first word read back from system memory is 0xcadefeca.

When the SHA_IODATARx registers are read, the hash result is organized in little-endian format, allowing system memory storage in the same format as the message.

Taking an example from the FIPS 180 specification Appendix B.1, the endian conversion can be observed.

For this example, the 512-bit message is:

0x61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018

and the expected SHA-256 result is:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

If the message has not already been stored in the system memory, the first step is to convert the input message to little-endian before writing to the SHA_IDATARx registers. This would result in a write of:

SHA_IDATAR0 = 0x80636261...... SHA_IDATAR15 = 0x18000000

The data in the output message digest registers, SHA_IODATARx, contain SHA_IODATAR0 = 0xbf1678ba... SHA_IODATAR7 = 0xad1500f2 which is the little-endian format of 0xba7816bf,..., 0xf20015ad.

Reading SHA_IODATAR0 to SHA_IODATAR1 and storing into a little-endian memory system forces hash results to be stored in the same format as the message.

When the output message is read, the user can convert back to big-endian for a resulting message value of:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

55.4.10 Security Features

55.4.10.1 Unspecified Register Access Detection

When an unspecified register access occurs, the URAD bit in the SHA_ISR is set. Its source is then reported in the Unspecified Register Access Type field (URAT). Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- SHA_IDATARx written during data processing in DMA mode
- SHA_IODATARx read during data processing
- SHA_MR written during data processing
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in the SHA_CR.

55.4.10.2 Register Write Protection

To prevent any single software error from corrupting SHA behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SHA Write Protection Mode register (SHA_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the SHA Write Protection Status register (SHA_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading SHA_WPSR.

The following register(s) can be write-protected when SHA_WPMR.WPEN is set:

- SHA Mode Register
- SHA Message Size Register
- SHA Bytes Count Register

The following register(s) can be write-protected when WPITEN is set:

- SHA Interrupt Enable Register
- SHA Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- SHA Control Register

55.4.10.3 Security and Safety Analysis and Reports

Several types of checks are performed when the SHA is enabled.

The peripheral clock of the SHA is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the SHA. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the SHA_WPSR.CGD flag is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the SHA is also monitored, and if an abnormal state is detected, the SHA_WPSR.SEQE flag is set. This flag is not set under normal operating conditions.

Software accesses to the SHA are monitored and if an incorrect access is performed, the SHA_WPSR.SWE flag is set. The type of incorrect/abnormal software access is reported in the SHA_WPSR.SWETYP field (see SHA Write Protection Status Register for details), e.g., reading the SHA_ODATARx when the SHA_ISR.DATRDY flag is cleared is an error. SHA_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The CGD, SEQE, SWE and WPVS flags are automatically cleared when SHA_WPSR is read.

If one of these flags is set, the SHA_ISR.SECE flag is set and can trigger an interrupt if SHA_IMR.SECE is ‘1’. SECE is cleared by reading SHA_ISR.

It is possible to configure an action to be performed by SHA as soon as an abnormal event detection occurs. If SHA_WPMR.ACTION > 0, a lock is performed. When a lock occurs, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see SHA_MR.SMOD).

A locked state of the SHA is unlocked as follows:

1. Read SHA_WPSR.
2. Disable the source of tamper if the tamper is enabled.
3. Write a ‘1’ to SHA_CR.UNLOCK.

It is possible to select the type of event that will lock the SHA in case of abnormal event detection. See SHA_WPMR.ACTION for details.

If SHA_MR.TMPLCK=1 and the tamper pin is active, the SHA is locked whatever the value of the field SHA_WPMR.ACTION.

55.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SHA_CR	31:24								UNLOCK
		23:16								
		15:8			WUIEHV	WUIHV				SWRST
		7:0				FIRST				START
0x04	SHA_MR	31:24	CHKCNT[3:0]						CHECK[1:0]	
		23:16								DUALBUFF
		15:8	TMPLCK				ALGO[3:0]			
		7:0	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
0x08 ... 0x0F	Reserved									
0x10	SHA_IER	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x14	SHA_IDR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x18	SHA_IMR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x1C	SHA_ISR	31:24								SECE
		23:16	CHKST[3:0]							CHECKF
		15:8		URAT[2:0]						URAD
		7:0				WRDY				DATRDY
0x20	SHA_MSR	31:24	MSGSIZE[31:24]							
		23:16	MSGSIZE[23:16]							
		15:8	MSGSIZE[15:8]							
		7:0	MSGSIZE[7:0]							
0x24 ... 0x2F	Reserved									
0x30	SHA_BCR	31:24	BYTCNT[31:24]							
		23:16	BYTCNT[23:16]							
		15:8	BYTCNT[15:8]							
		7:0	BYTCNT[7:0]							
0x34 ... 0x3F	Reserved									
0x40	SHA_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
...										
0x7C	SHA_IDATAR15	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x80	SHA_IODATAR0	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							
...										

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xBC	SHA_IODATAR15	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							
0xC0 ... 0xE3	Reserved									
0xE4	SHA_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
0xE8	SHA_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

55.5.1 SHA Control Register

Name: SHA_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			WUIEHV	WUIHV				SWRST
Access			W	W				W
Reset			–	–				–

Bit	7	6	5	4	3	2	1	0
				FIRST				START
Access				W				W
Reset				–				–

Bit 24 – UNLOCK Unlock Processing
SHA_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if SHA_WPMR.ACTION > 0.

Bit 13 – WUIEHV Write User Initial or Expected Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR1).

Bit 12 – WUIHV Write User Initial Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR0).

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the SHA. A software-triggered hardware reset of the SHA interface is performed.

Bit 4 – FIRST First Block of a Message

Value	Description
0	No effect.
1	Indicates that the next block to process is the first one of a message or the first block of a fragment of a message.

Bit 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual hash algorithm process.

55.5.2 SHA Mode Register

Name: SHA_MR
Offset: 0x04
Reset: 0x0000100
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHKCNT[3:0]						CHECK[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
								DUALBUFF
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TMPLCK				ALGO[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	1

Bit	7	6	5	4	3	2	1	0
	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 31:28 – CHKCNT[3:0] Check Counter
Number of 32-bit words to check. The value 0 indicates that the number of words to compare will be based on the algorithm selected (5 words for SHA1, 7 words for SHA224, 8 words for SHA256, 12 words for SHA384, 16 words for SHA512).

Bits 25:24 – CHECK[1:0] Hash Check
Values not listed in table must be considered as “reserved”.

Value	Name	Description
0	NO_CHECK	No check is performed.
1	CHECK_EHV	Check is performed with expected hash stored in internal expected hash value registers.
2	CHECK_MESSAGE	Check is performed with expected hash provided after the message.

Bit 16 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	SHA_IDATARx and SHA_IODATARx cannot be written during processing of previous block.
1	ACTIVE	SHA_IDATARx and SHA_IODATARx can be written during processing of previous block when SMOD value = 2. It speeds up the overall runtime of large files.

Bit 15 – TMPLCK Tamper Lock Enable

Value	Description
0	A tamper event has no effect.
1	A tamper event locks the SHA until the tamper root cause is cleared and SHA_CR.UNLOCK is written to 1.

Bits 11:8 – ALGO[3:0] SHA Algorithm
Values not listed in the table must be considered as “reserved”.

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
2	SHA384	SHA384 algorithm processed
3	SHA512	SHA512 algorithm processed
4	SHA224	SHA224 algorithm processed
5	SHA512_224	SHA512/224 algorithm processed
6	SHA512_256	SHA512/256 algorithm processed
8	HMAC_SHA1	HMAC algorithm with SHA1 Hash processed
9	HMAC_SHA256	HMAC algorithm with SHA256 Hash processed
10	HMAC_SHA384	HMAC algorithm with SHA384 Hash processed
11	HMAC_SHA512	HMAC algorithm with SHA512 Hash processed
12	HMAC_SHA224	HMAC algorithm with SHA224 Hash processed
13	HMAC_SHA512_224	HMAC algorithm with SHA512/224 Hash processed
14	HMAC_SHA512_256	HMAC algorithm with SHA512/256 Hash processed

Bit 7 – BPE Block Processing End
When SMOD=2 and ALGO<5, the SHA_ISR.DATRDY flag rises when each block has been processed.
When SMOD=2 and ALGO>7, the SHA_ISR.DATRDY rises when all blocks except the last one have been processed.

Value	Description
0	BPE must be cleared when a DMA transfers data. When SMOD=2, SHA_ISR.DATRDY flag rises only when the SHA or HMAC processing cycle has completed. No intermediate block processing is reported.
1	When processing small messages, data transfer by software can improve performance compared to DMA. In this case, BPE can be written to 1, forcing the SHA_ISR.DATRDY to rise when a data must be loaded into SHA_IDATARx.

Bit 6 – UIEHV User Initial or Expected Hash Value Registers

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS 180 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 1 (IR1). If HMAC is configured, UIEHV has no effect (i.e. IR1 is always selected).

Bit 5 – UIHV User Initial Hash Values

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS 180 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 0 (IR0). If HMAC is configured, UIHV has no effect (i.e. IR0 is selected).

Bit 4 – PROCDLY Processing Delay
When SHA1 algorithm is processed, runtime period is either 85 or 209 clock cycles.
When SHA256 or SHA224 algorithm is processed, runtime period is either 72 or 194 clock cycles.
When SHA384 or SHA512 algorithm is processed, runtime period is either 88 or 209 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one
1	LONGEST	SHA processing runtime is the longest one (reduces the SHA bandwidth requirement, reduces the system bus overload)

Bit 3 – AOE Always On Enable

Value	Description
0	The SHA operates in functional operating modes.
1	As soon as a START command is written, the SHA processes dummy calculations until AOE=0, without software intervention. This can be used to create an additional current consumption when AES is used to encrypt/decrypt.

Bits 1:0 – SMOD[1:0] Start Mode
Values not listed in the table must be considered as “reserved”.
If a DMA transfer is used, configure the SMOD value to 2. See DMA Mode for details.

Value	Name	Description
0	MANUAL_START	Manual mode
1	AUTO_START	Auto mode
2	IDATAR0_START	SHA_IDATAR0 access only mode (mandatory when DMA is used)

55.5.3 SHA Interrupt Enable Register

Name: SHA_IER
Offset: 0x10
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 24 – SECE Security and/or Safety Event Interrupt Enable

Bit 16 – CHECKF Check Done Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

55.5.4 SHA Interrupt Disable Register

Name: SHA_IDR
Offset: 0x14
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 24 – SECE Security and/or Safety Event Interrupt Disable

Bit 16 – CHECKF Check Done Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

55.5.5 SHA Interrupt Mask Register

Name: SHA_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 24 – SECE Security and/or Safety Event Interrupt Mask

Bit 16 – CHECKF Check Done Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

55.5.6 SHA Interrupt Status Register

Name: SHA_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	CHKST[3:0]							CHECKF
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
		URAT[2:0]						URAD
Access		R	R	R				R
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
				WRDY				DATRDY
Access				R				R
Reset				0				0

Bit 24 – SECE Security and/or Safety Event

Value	Description
0	There is no report in SHA_WPSR.
1	There is a Security and/or Safety Event reported in SHA_WPSR.

Bits 23:20 – CHKST[3:0] Check Status (cleared by writing SHA_CR.START or SHA_CR.SWRST or by reading SHA_IODATARx)
Value 5 indicates identical hash values (expected hash = hash result). Any other value indicates different hash values.

Bit 16 – CHECKF Check Done Status (cleared by writing SHA_CR.START or SHA_CR.SWRST or by reading SHA_IODATARx)

Value	Description
0	Hash check has not been computed.
1	Hash check has been computed, status is available in the CHKST bits.

Bits 14:12 – URAT[2:0] Unspecified Register Access Type (cleared by writing a 1 to SWRST bit in SHA_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name
0	SHA_IDATAR0 to SHA_IDATAR15 written during data processing in DMA mode (URAD = 1 and URAT = 0 can occur only if DUALBUFF is cleared in SHA_MR)
1	Output Data Register read during data processing
2	SHA_MR written during data processing
3	Write-only register read access

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing a 1 to SHA_CR.SWRST)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 4 – WRDY Input Data Register Write Ready

Value	Description
0	SHA_IDATAR0 cannot be written
1	SHA_IDATAR0 can be written

Bit 0 – DATRDY Data Ready (cleared by writing a 1 to bit SWRST or START in SHA_CR, or by reading SHA_IODATARx)

Value	Description
0	Output data is not valid.
1	512/1024-bit block process is completed. DATRDY is cleared when one of the following conditions is met: • Bit START in SHA_CR is set. • Bit SWRST in SHA_CR is set. • The hash result is read.

55.5.7 SHA Message Size Register

Name: SHA_MSR
Offset: 0x20
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MSGSIZE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MSGSIZE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MSGSIZE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MSGSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSGSIZE[31:0] Message Size
The size in bytes of the message. When MSGSIZE differs from 0, the SHA appends the corresponding value converted in bits after the padding section, as described in the FIPS180 specification.
To disable automatic padding, MSGSIZE field must be written to 0.
Note: SHA_MSR is a 32-bit register, thus the automatic padding capability is limited to messages of less than 4 gigabytes. For messages greater than 4 gigabytes, padding must be performed by the software.

55.5.8 SHA Bytes Count Register

Name: SHA_BCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BYTCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BYTCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BYTCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BYTCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BYTCNT[31:0] Remaining Byte Count Before Auto Padding
When the hash processing starts from the beginning of a message (without preprocessed hash part), BYTCNT must be written with the same value as MSGSIZE. If a part of the message has been already hashed and the hash does not start from the beginning, BYTCNT must be configured with the number of bytes remaining to process before the padding section.
When read, provides the size in bytes of the message remaining to be written before the automatic padding starts. BYTCNT is automatically updated each time a write occurs in SHA_IDATARx and SHA_IODATARx.
When BYTCNT reaches 0, the MSGSIZE is converted into a bit count and appended at the end of the message after the padding, as described in the FIPS 180 specification.
To disable automatic padding, the MSGSIZE and BYTCNT fields must be written to 0.

55.5.9 SHA Input Data Register x

Name: SHA_IDATARx
Offset: 0x40 + x*0x04 [x=0..15]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data
32-bit Input Data registers load the data block used for hash processing.
These registers are write-only to prevent reading of input data by another application.
SHA_IDATAR0 corresponds to the first word of the block, SHA_IDATAR15 to the last word of the last block in case SHA algorithm is set to SHA1, SHA224, SHA256, or SHA_IODATAR15 to the last word of the block if SHA algorithm is SHA384 or SHA512 (see SHA Input/Output Data Register x).
SHA_IDATARx can be also written to configure the hash result of the previous fragment of a message when starting the processing of the next fragment when the SHA has processed another message in between fragments.

55.5.10 SHA Input/Output Data Register x

Name: SHA_IODATARx
Offset: 0x80 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	IODATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	IODATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	IODATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	IODATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IODATA[31:0] Input/Output Data
These registers can be used to read the resulting message digest and to write the second part of the message block when the SHA algorithm is SHA-384 or SHA-512.
SHA_IODATAR0 to SHA_IODATAR15 can be written or read but reading these offsets does not return the content of corresponding parts (words) of the message block. Only results from SHA calculation can be read through these registers.
When SHA processing is in progress, these registers return 0x0000.
SHA_IODATAR0 corresponds to the first word of the message digest; SHA_IODATAR4 to the last one in SHA1 mode, SHA_IODATAR6 in SHA224, SHA_IODATAR7 in SHA256, SHA_IODATAR11 in SHA384 or SHA_IODATAR15 in SHA512.
When SHA224 is selected, the content of SHA_IODATAR7 must be ignored.
When SHA384 is selected, the content of SHA_IODATAR12 to SHA_IODATAR15 must be ignored.

55.5.11 SHA Write Protection Mode Register

Name: SHA_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534841	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 6:5 – ACTION[1:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of WPVS,CGD,SEQE, or SWE flag is set.
1	LOCK_WPVS_SWE	If a processing is in progress when the SHA_WPSR.WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the SHA_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the SHA_WPSR.WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in SHA_WPSR.WPVSRC and the last software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in SHA_WPSR.WPVSRC and only the first software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

55.5.12 SHA Write Protection Status Register

Name: SHA_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the SHA is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	SHA is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	SHA is locked and a start command with SHA_CR.START has been performed.
4	AUTO_START	SHA is locked and a tentative automatic start has been performed by writing input data registers (SHA_MR.SMOD>0).
5	BAD_START	SHA is not locked and a start command with SHA_CR.START has been performed whereas Start mode is automatic (SHA_MR.SMOD>0)

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SHA_WPSR.
1	A software error has occurred since the last read of SHA_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of SHA_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of SHA_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of SHA_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of SHA_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of SHA_WPSR.
1	A write protect violation has occurred since the last read of SHA_WPSR. The address offset of the violated register is reported into field WPVSRC.

56. Triple Data Encryption Standard (TDES)

56.1 Description

The Triple Data Encryption Standard (TDES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 46-3 specification.

The TDES supports the four different confidentiality modes of operation (ECB, CBC, OFB and CFB), specified in the FIPS (Federal Information Processing Standard) Publication 81 and is compatible with the Peripheral Data Controller channels for all of these modes, minimizing processor intervention for large buffer transfers.

The TDES key can be either loaded by the software or loaded in an invisible manner from the software.

The software can write up to three 64-bit keys, each stored in two 32-bit write-only registers, i.e., Key x Word registers, TDES_KEYxWR0 and TDES_KEYxWR1. For a software-invisible key transfer, the Private Key bus accesses the Private Key internal registers from the TRNG or OTPC. The PKRS bit in the Mode register selects either TDES_KEYxWR0/TDES_KEYxWR1 or the Private Key internal registers.

The input data (and initialization vector for some modes) are stored in two corresponding 32-bit write-only registers:

- Input Data registers, TDES_IDATAR0 and TDES_IDATAR1
- Initialization Vector registers, TDES_IVR0 and TDES_IVR1

As soon as the initialization vector, the input data and the keys are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data is ready to be read out on the two 32-bit Output Data registers (TDES_ODATARx) or through the DMA channels.

56.2 Embedded Characteristics

- Supports Single Data Encryption Standard (DES) and Triple Data Encryption Standard (TDES)
- Compliant with FIPS Publication 46-3, Data Encryption Standard (DES)
- 64-bit Cryptographic Key for TDES
- Two-key or Three-key Algorithms for TDES
- 18 Clock Cycles Encryption/Decryption Processing Time for DES
- 50 Clock Cycles Encryption/Decryption Processing Time for TDES
- Supports eXtended Tiny Encryption Algorithm (XTEA)
- 128-bit key for XTEA and Programmable Round Number up to 64
- Supports the Four Standard Modes of Operation specified in the FIPS Publication 81, DES Modes of Operation
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
- 8-, 16-, 32- and 64-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allowing Optimized Message (Data) Authentication Code (MAC) Generation
- Abnormal Software Access Reports and Automatic Lock
- Abnormal Internal Sequence Detection and Automatic Lock
- Register Write Protection
- Temporary Secured Storage for Keys
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software
- Connection to DMA Optimizes Data Transfers for all Operating Modes

56.3 Product Dependencies

56.3.1 Power Management

The TDES may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the TDES clock.

56.3.2 Interrupt Sources

The TDES interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by TDES_IMR and TDES_ISR, whereas the interrupt line for safety/security functions is driven by TDES_WPSR flags. If one of the flags WPVS, CGD, SEQE or SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TDES.

56.4 Functional Description

The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDES) specify FIPS-approved cryptographic algorithms that can be used to protect electronic data. TDES_MR.TDES is used to select either the single DES or the Triple DES mode.

Encryption (enciphering) converts data to an unintelligible form called ciphertext. Decrypting (deciphering) the ciphertext converts the data back into its original form, called plaintext. TDES_MR.CIPHER is used to choose between encryption and decryption.

A DES is capable of using cryptographic keys of 64 bits to encrypt and decrypt data in blocks of 64 bits. This 64-bit key is defined in the Key 1 registers (TDES_KEY1WRx or Private Key internal registers, only writable from the Private Key bus).

A TDES key consists of three DES keys, which is also referred to as a key bundle. These three 64-bit keys are defined, respectively, in the Key 1, 2 and 3 Registers (TDES_KEY1WRy, TDES_KEY2WRy and TDES_KEY3WRy or the Private Key internal registers). In Triple DES mode (TDESMOD = 1 in TDES_MR), TDES_MR.KEYMOD is used to choose between a two- and a three-key algorithm, as summarized in the table below.

Table 56-1. TDES Algorithms Summary

Algorithm	Mode	Data Processing Sequence Steps		
		First	Second	Third
Three-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 3
	Decryption	Decryption with Key 3	Encryption with Key 2	Decryption with Key 1
Two-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 1
	Decryption	Decryption with Key 1	Encryption with Key 2	Decryption with Key 1

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 64-bit data block called the initialization vector (IV), which must be set in TDES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message.

The XTEA algorithm can be used instead of DES/TDES by configuring TDES_MR.TDESMOD with the appropriate value 0x2. An XTEA key consists of a 128-bit key. They are defined in the Key 1 and 2 Registers.

The number of rounds of XTEA is defined in TDES_XTEA_RNDR and can be programmed up to 64 (1 round = 2 Feistel network rounds).

All the start and operating modes of the TDES algorithm can be applied to the XTEA algorithm.

56.4.1 Operating Modes

The TDES supports the following operating modes:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining
- OFB—Output Feedback
- CFB—Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)

The data pre-processing, post-processing and data chaining for each mode are automatically performed. Refer to the FIPS Publication 81 for more complete information.

These modes are selected by setting TDES_MR.OPMOD.

In CFB mode, four data sizes are possible (8, 16, 32 and 64 bits), configurable in TDES_MR.CFBS (see TDES Mode Register).

56.4.2 Temporary Secured Storage for Keys

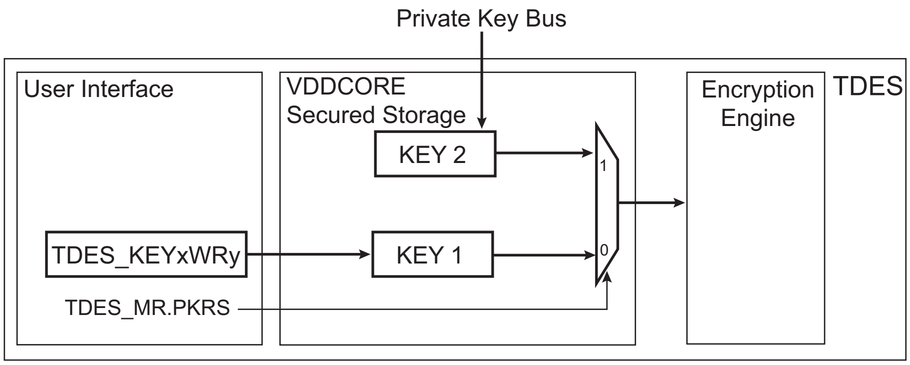
The TDES provides secure storage for two sets of three 64-bit keys. The storage is available while VDDCORE voltage is supplied.

The keys can be only written in TDES internal registers and are not readable. Moreover, the internal registers holding the keys are buried in the overall product logic area during the physical implementation.

One set of keys can be loaded by software by writing the Key Word registers (TDES_KEYxWRy).

One key can be loaded by Private Key bus only.

Figure 56-1. Temporary Secured Storage for Keys



56.4.3 Start Modes

TDES_MR.SMOD selects the Encryption (or Decryption) start mode.

56.4.3.1 Manual Mode

The sequence is as follows:

1. Write TDES_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 64-bit key(s) in TDES_KEYxWRy or the Private Key internal register, depending on whether one, two or three keys are required.
3. Write the initialization vector (or counter) in TDES_IVRx.
   Note: TDES_IVRx concern all modes except ECB.
4. Set DATRDY (Data Ready) in the TDES Interrupt Enable register (TDES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized TDES_IDATARx (see the table below).
   Note: In 32-, 16- and 8-bit CFB modes, writing to TDES_IDATAR1 is not allowed and may lead to processing errors.
6. Set the START bit in the TDES Control Register (TDES_CR) to begin the encryption or decryption process.

7. When the processing completes, DATRDY in the TDES Interrupt Status register (TDES_ISR) rises. If an interrupt has been enabled by setting TDES_IER.DATRDY, the interrupt line of the TDES is activated.
8. When the software reads a TDES_ODATARx, TDES_IER.DATRDY is automatically cleared.

Table 56-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
CFB 64-bit	All
CFB 32-bit	TDES_IDATAR0
CFB 16-bit	TDES_IDATAR0
CFB 8-bit	TDES_IDATAR0

56.4.3.2 Auto Mode

The Auto Mode is similar to the Manual Mode, except that as soon as the correct number of TDES_IDATARx is written, processing is automatically started without any action in TDES_CR.

56.4.3.3 DMA Mode

The DMA Controller can be used in association with the TDES to perform an encryption/decryption of a buffer without any action by the software during processing.

TDES_MR.SMOD must be set to 2 and the DMA must be configured with non-incremental addresses.

For all operating modes except CBC-MAC (TDES_MR.LOD=1), 2 DMA channels must be programmed (transmit and receive). In CBC-MAC, only 1 transmit channel must be programmed.

The start address of any transfer descriptor must be set in TDES_IDATAR0.

The DMA chunk size configuration depends on the TDES mode of operation and is listed in the table below.

When writing data to TDES with the first DMA channel, data will be fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the TDES with the second DMA channel, the source data is the data read from TDES and data destination is the memory buffer. In this case, source data size depends on the TDES mode of operation and is listed in the table below.

Table 56-3. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	1	Word
CBC	1	Word
OFB	1	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte

56.4.4 Last Output Data Mode (CBC-MAC)

This mode is used to generate cryptographic checksums on data (MAC) using a CBC-MAC or a CFB encryption algorithm (refer to *FIPS Publication 81 Appendix F*).

The CMAC algorithm is a variant of CBC-MAC with post-processing requiring one-block encryption in ECB mode. Thus CBC-MAC is useful to accelerate CMAC.

After each end of encryption/decryption, the output data is available either on the output data registers for Manual and Auto modes or at the address specified in the receive buffer pointer for DMA mode (See Last Output Data Mode Behavior versus Start Modes).

TDES_MR.LOD can be used to retrieve only the last data of several encryption/decryption processes.

This data is only available in TDES_ODATARx.

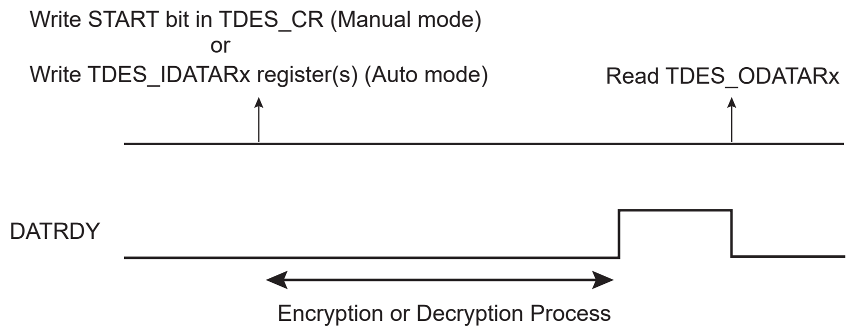
Therefore, there is no need to define a read buffer in DMA mode.

56.4.4.1 Manual and Auto Modes

56.4.4.1.1 TDES_MR.LOD = 0

The DATRDY flag is cleared when at least one TDES_ODATARx is read. See the figure below.

Figure 56-2. Manual and Auto Modes with LOD = 0

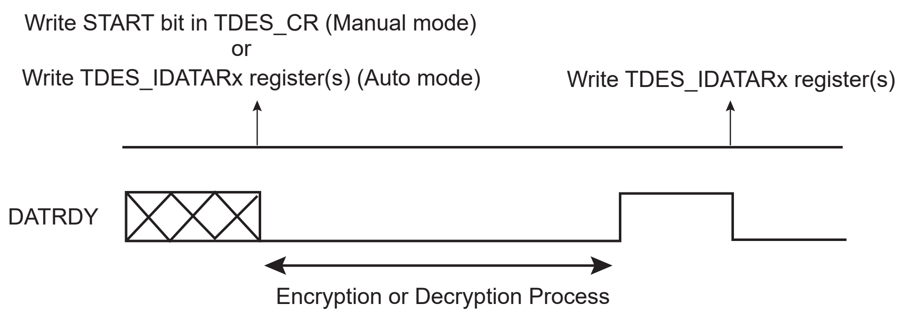


If the user does not want to read TDES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user will not be informed of the end of the encryptions/decryptions that follow.

56.4.4.1.2 TDES_MR.LOD = 1

The DATRDY flag is cleared when at least one TDES_IDATARx is written, before the start of a new transfer. See the figure below. No further TDES_ODATARx reads are necessary between consecutive encryptions/decryptions.

Figure 56-3. Manual and Auto Modes with LOD = 1



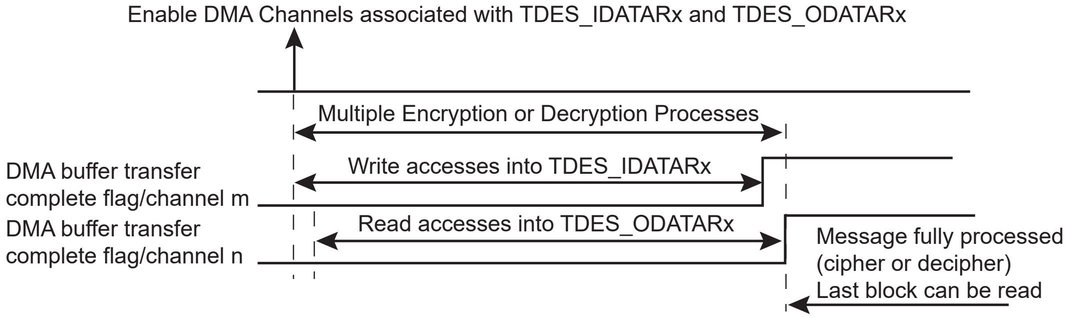
56.4.4.2 DMA Mode

56.4.4.2.1 TDES_MR.LOD = 0

This mode may be used for all TDES operating modes except CBC-MAC where LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to TDES_ODATARx (see the figure below). Two DMA channels are required: one for writing message blocks to TDES_IDATARx and one to obtain the result from TDES_ODATARx.

Figure 56-4. DMA Transfer with LOD = 0



56.4.4.2.2 TDES_MR.LOD = 1

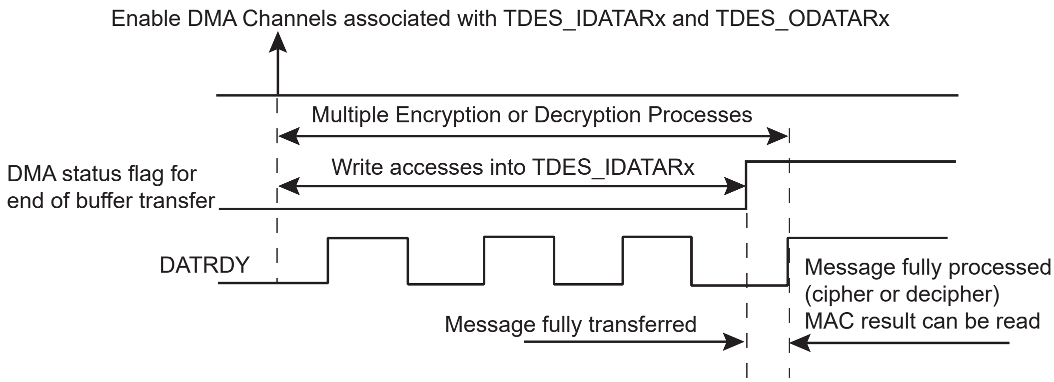
This mode is optimized to process the TDES CBC-MAC operating mode.

The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see the figure below).

The DMA receive channel must not be used. Prior to reading the CBC-MAC result, TDES_MR.SMOD must be written to 0. To restart a CBC-MAC on a new buffer, TDES_MR.SMOD must be written to 2.

The output data is only available on TDES_ODATARx.

Figure 56-5. DMA Transfer with LOD = 1



The table below summarizes the different cases.

Table 56-4. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	LOD = 0	LOD = 1	LOD = 0	LOD = 1
DATRDY Flag Clearing Condition (1)	At least one TDES_ODATARx must be read	At least one TDES_IDATARx must be written	Not used	Managed by the DMA
End of Encryption/ Decryption	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then TDES DATRDY flag
Encrypted/ Decrypted Data Result Location	In TDES_ODATARx	In TDES_ODATARx	Not available	In TDES_ODATARx

Note: Depending on the mode, there are other ways of clearing the DATRDY flag. See TDES Interrupt Status Register.

⚠WARNING In DMA mode, reading to TDES_ODATARx before the last data transfer may lead to unpredictable results.

56.4.5 Security Features

56.4.5.1 Private Key Bus

The TDES provides secure key transfer that requires a transfer command only, thus avoiding any manipulation of the key by software.

The TDES features a set of Private Key internal registers that can be accessed only through the dedicated Private Key bus from the TRNG or OTPC.

The Private Key internal registers cannot be read from any peripheral or from software.

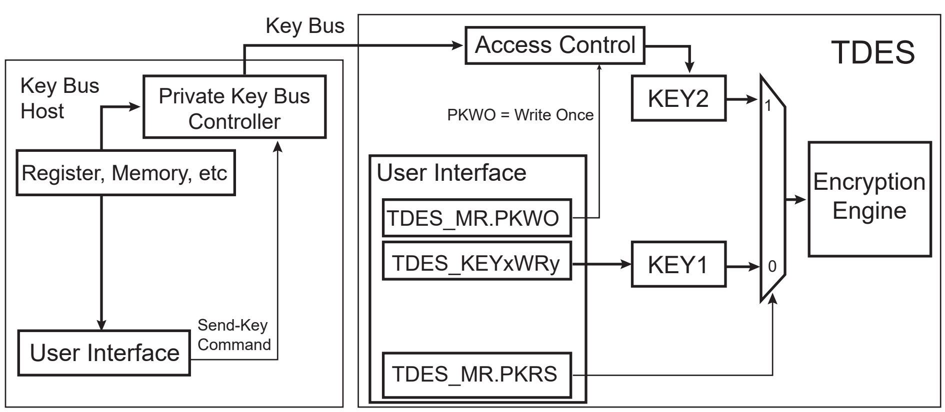
The TDES key used by the encryption/decryption engine is either the Private Key internal registers content or the internal key registers loaded via the TDES_KEYxWRy.

To select the Private Key internal registers as the source of the TDES key, TDES_MR.PKRS must be written to '1'.

To write the Private Key internal registers, the software must:

1. Write a '1' in TDES_MR.PKRS.
2. Trigger the key transfer over the Private Key bus from the KEY_BUS_MASTERS key bus host.
3. Wait for completion of the transfer signaled in the KEY_BUS_MASTERS status register.
4. Check for any access violation in TDES_WPSR.PKRPVS.

Figure 56-6. Key Selection



While TDES_MR.PKWO=0, it is possible to write the Private Key internal registers as many times as required.

As soon as the bit TDES_MR.PKWO=1, the next write sequence on Private Key internal registers is the last one. Any additional write sequence in the Private Key internal registers has no effect, thus providing write-protection of these registers. A hardware reset is the only way to exit from the write-protected state.

56.4.5.2 Unspecified Register Access Detection

When an unspecified register access occurs, TDES_ISR.URAD is set. Its source is then reported in TDES_ISR.URAT. Only the last unspecified register access is available through TDES_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- TDES_IDATARx written during the data processing in DMA mode
- TDES_ODATARx read during the data processing
- TDES_MR written during the data processing
- Write-only register read access

URAD and URAT can only be reset by TDES_CR.SWRST.

56.4.5.3 Clearing Key on Tamper Event

On a tamper detection event on WKUP1..8 pins, an immediate clear of the key (internal registers) can be performed if TDES_MR.TAMPCLR=1. For configuration details, refer to section Real-Time Clock (RTC).

56.4.5.4 Register Write Protection

To prevent any single software error from corrupting TDES behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TDES Write Protection Mode Register (TDES_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TDES Write Protection Status Register (TDES_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically cleared by reading TDES_WPSR.

The following register can be write-protected when WPEN is set:

- TDES Mode Register
- TDES Key 1 Word Register x
- TDES Key 2 Word Register x
- TDES Key 3 Word Register x
- TDES Initialization Vector Register x
- TDES XTEA Rounds Register

The following registers can be write-protected when WPITEN is set:

- TDES Interrupt Enable Register
- TDES Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TDES Control Register

56.4.5.5 Security and Safety Analysis and Reports

Several types of checks are performed when the TDES is enabled.

The peripheral clock of the TDES is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TDES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TDES_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TDES is also monitored and if an abnormal state is detected, the flag TDES_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TDES are monitored and if an incorrect access is performed, the flag TDES_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the TDES_WPSR.SWETYP field (see TDES Write Protection Status Register for details). e.g., writing the TDES_ODATARx is an error, as well as reading the TDES_IDATARx, when the TDES_ISR.DATRDY flag is cleared. TDES_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TDES_WPSR is read.

If one of these flags is set, the flag TDES_ISR.SECE is set and can trigger an interrupt if the TDES_IMR.SECE bit is ‘1’. SECE is cleared by reading TDES_ISR.

It is possible to configure an action to be performed by the TDES as soon as an abnormal event detection occurs. If the field TDES_WPMR.ACTION is greater than 0, either a lock is performed or a lock and immediate clear of TDES_KEYxWRy. If a lock is performed, the current processing is ended normally but any new processing is not performed regardless of the start mode of operation (see TDES_MR.SMOD).

A locked state of the TDES is unlocked as follows:

1. Read the TDES_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to TDES_CR.UNLOCK.

It is possible to select the type of event that will lock the TDES in case of abnormal event detection. See TDES_WPMR.ACTION for details.

If the TDES_MR.TMPCLR=1 and the tamper pin is active, the TDES is locked.

56.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TDES_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0								START
0x04	TDES_MR	31:24	TAMPCLR							
		23:16							CFBS[1:0]	
		15:8	LOD		OPMOD[1:0]				SMOD[1:0]	
		7:0	PKRS	PKWO		KEYMOD		TDESMOD[1:0]		CIPHER
0x08 ... 0x0F	Reserved									
0x10	TDES_IER	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x14	TDES_IDR	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x18	TDES_IMR	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x1C	TDES_ISR	31:24								
		23:16								SECE
		15:8			URAT[1:0]					URAD
		7:0								DATRDY
0x20	TDES_KEY1WR0	31:24	KEY1W[31:24]							
		23:16	KEY1W[23:16]							
		15:8	KEY1W[15:8]							
		7:0	KEY1W[7:0]							
0x24	TDES_KEY1WR1	31:24	KEY1W[31:24]							
		23:16	KEY1W[23:16]							
		15:8	KEY1W[15:8]							
		7:0	KEY1W[7:0]							
0x28	TDES_KEY2WR0	31:24	KEY2W[31:24]							
		23:16	KEY2W[23:16]							
		15:8	KEY2W[15:8]							
		7:0	KEY2W[7:0]							
0x2C	TDES_KEY2WR1	31:24	KEY2W[31:24]							
		23:16	KEY2W[23:16]							
		15:8	KEY2W[15:8]							
		7:0	KEY2W[7:0]							
0x30	TDES_KEY3WR0	31:24	KEY3W[31:24]							
		23:16	KEY3W[23:16]							
		15:8	KEY3W[15:8]							
		7:0	KEY3W[7:0]							
0x34	TDES_KEY3WR1	31:24	KEY3W[31:24]							
		23:16	KEY3W[23:16]							
		15:8	KEY3W[15:8]							
		7:0	KEY3W[7:0]							
0x38 ... 0x3F	Reserved									

...........continued										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	TDES_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x44	TDES_IDATAR1	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x48 ... 0x4F	Reserved									
0x50	TDES_ODATAR0	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54	TDES_ODATAR1	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x58 ... 0x5F	Reserved									
0x60	TDES_IVR0	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x64	TDES_IVR1	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x68 ... 0x6F	Reserved									
0x70	TDES_XTEA_RNDR	31:24								
		23:16								
		15:8								
		7:0			XTEA_RNDS[5:0]					
0x74 ... 0xE3	Reserved									
0xE4	TDES_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	TDES_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS

56.5.1 TDES Control Register

Name: TDES_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								–

Bit 24 – UNLOCK Unlock Processing

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if TDES_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the TDES. A software-triggered reset of the TDES interface is performed.

Bit 0 – START Start Processing

Value	Description
0	No effect
1	Starts Manual encryption/decryption process.

56.5.2 TDES Mode Register

Name: TDES_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
							CFBS[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	LOD		OPMOD[1:0]				SMOD[1:0]	
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bit	7	6	5	4	3	2	1	0
	PKRS	PKWO		KEYMOD		TDESMOD[1:0]		CIPHER
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	0	0		0		0	0	0

Bit 31 – TAMPCLR Tamper Pin Clear Key Enable

Value	Description
0	A tamper detection event has no effect on TDES_KEYxWRy.
1	A tamper detection event immediately clears TDES_KEYxWRy.

Bits 17:16 – CFBS[1:0] Cipher Feedback Data Size

Value	Name	Description
0	SIZE_64BIT	64-bit
1	SIZE_32BIT	32-bit
2	SIZE_16BIT	16-bit
3	SIZE_8BIT	8-bit

Bit 15 – LOD Last Output Data Mode

⚠WARNING In DMA mode, reading to TDES_ODATARx before the last data encryption/decryption process may lead to unpredictable result.

Value	Description
0	No effect. After each end of encryption/decryption, the output data is available either on TDES_ODATARx (Manual and Auto modes) . In Manual and Auto modes, the DATRDY flag is cleared when at least one of the TDES_ODATARx is read.

Value	Description
1	The DATRDY flag is cleared when at least one of the Input Data Registers is written. No further TDES_ODATARx reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

Bits 13:12 – OPMOD[1:0] Operating Mode
For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.

Value	Name	Description
0	ECB	Electronic Code Book mode
1	CBC	Cipher Block Chaining mode
2	OFB	Output Feedback mode
3	CFB	Cipher Feedback mode

Bits 9:8 – SMOD[1:0] Start Mode
If a DMA transfer is used, 0x2 must be configured. See DMA Mode for more details.

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	TDES_IDATAR0 accesses only Auto Mode

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The keys used by the TDES are in the TDES_KEY1WRx, TDES_KEY2WRx and TDES_KEY3WRx registers.
1	The keys used by the TDES are the in the Private Key internal registers written through the Private Key bus.

Bit 6 – PKWO Private Key Write Once
Once PKWO is set to '1', only a hardware reset sets this bit to '0' internally. Writing it to '0' with a register access has no impact (although the field will be read to value '0').

Value	Description
0	The Private Key internal register can be written multiple times through the Private Key bus.
1	The Private Key internal register can be written only once through the Private Key bus until hardware reset.

Bit 4 – KEYMOD Key Mode

Value	Description
0	Three-key algorithm is selected.
1	Two-key algorithm is selected. There is no need to write TDES_KEY3WRy (or Private Pey internal registers with more than 128 bits).

Bits 2:1 – TDESMOD[1:0] ALGORITHM Mode
Values which are not listed in the table must be considered as "reserved".

Value	Name	Description
0	SINGLE_DES	Single DES processing using TDES_KEY1WRy.
1	TRIPLE_DES	Triple DES processing using TDES_KEY1WRy, TDES_KEY2WRy and TDES_KEY3WRy .
2	XTEA	XTEA processing using TDES_KEY1WRy and TDES_KEY2WRy.

Bit 0 – CIPHER Processing Mode

Value	Name	Description
0	DECRYPT	Decrypts data.
1	ENCRYPT	Encrypts data.

56.5.3 TDES Interrupt Enable Register

Name: TDES_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TDES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SECE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 16 – SECE Security and/or Safety Event Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

56.5.4 TDES Interrupt Disable Register

Name: TDES_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TDES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								SECE
Access								W
Reset								–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
								URAD
Access								W
Reset								–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								DATRDY
Access								W
Reset								–

Bit 16 – SECE Security and/or Safety Event Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

56.5.5 TDES Interrupt Mask Register

Name: TDES_IMR
Offset: 0x18
Reset: 0x000000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								SECE
Access								R
Reset								0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
								URAD
Access								R
Reset								0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								DATRDY
Access								R
Reset								0

Bit 16 – SECE Security and/or Safety Event Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

56.5.6 TDES Interrupt Status Register

Name: TDES_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SECE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
			URAT[1:0]					URAD
Access			R	R				R
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 16 – SECE Security and/or Safety Event Interrupt Mask

Value	Description
0	There is no security report in TDES_WPSR.
1	One security flag is set in TDES_WPSR.

Bits 13:12 – URAT[1:0] Unspecified Register Access (cleared by setting bit TDES_CR.SWRST)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name	Description
0	IDR_WR_PROCESSING	TDES_IDATAR written during data processing when SMOD = 0x2 mode.
1	ODR_RD_PROCESSING	TDES_ODATAR read during data processing.
2	MR_WR_PROCESSING	TDES_MR written during data processing.
3	WOR_RD_ACCESS	Write-only register read access.

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by setting TDES_CR.SWRST)

Value	Description
0	No unspecified register access has been detected since the last write of TDES_CR.SWRST.
1	At least one unspecified register access has been detected since the last write of TDES_CR.SWRST.

Bit 0 – DATRDY Data Ready (cleared by setting TDES_CR.START or TDES_CR.SWRST, or by reading TDES_ODATARx)
If TDES_MR.LOD = 1: In Manual and Auto modes, the DATRDY flag can also be cleared by writing at least one TDES_IDATARx.

Value	Description
0	Output data is not valid.
1	Encryption or decryption process is completed.

56.5.7 TDES Key 1 Word Register y

Name: TDES_KEY1WRy
Offset: 0x20 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY1W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	KEY1W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	KEY1W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	KEY1W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY1W[31:0] Key 1 Word
The two 32-bit Key 1 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.
TDES_KEY1WR0.KEY1W refers to the first word of the key and TDES_KEY1WR1.KEY1W to the last one.
These registers are write-only to prevent the key from being read by another application.
In XTEA mode, the key is defined on 128 bits. These registers contain the 64 LSB bits of the encryption/decryption key.
TDES_KEY1WRy registers are not used if the Private Key internal register is selected instead by writing a 1 to TDES_MR.PKRS.

56.5.8 TDES Key 2 Word Register y

Name: TDES_KEY2WRy
Offset: 0x28 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY2W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY2W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY2W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY2W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY2W[31:0] Key 2 Word
The two 32-bit Key 2 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.
TDES_KEY2WR0.KEY2W refers to the first word of the key and TDES_KEY2W1.KEY2W to the last one.
These registers are write-only to prevent the key from being read by another application.
TDES_KEY2WRx registers are not used in DES mode.
In XTEA mode, the key is defined on 128 bits. These registers contain the 64 MSB bits of the encryption/decryption key.
TDES_KEY2WRy registers are not used if the Private Key internal register is selected instead by writing a 1 to TDES_MR.PKRS.

56.5.9 TDES Key 3 Word Register y

Name: TDES_KEY3WRy
Offset: 0x30 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY3W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY3W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY3W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY3W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY3W[31:0] Key 3 Word
The two 32-bit Key 3 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.
TDES_KEY3WR0.KEY3W refers to the first word of the key and TDES_KEY3WR1.KEY3W to the last one.
These registers are write-only to prevent the key from being read by another application.
TDES_KEY3WRx registers are not used in DES mode, TDES with two-key algorithm selected and XTEA mode.
TDES_KEY3WRy registers are not used if the Private Key internal register is selected by writing a 1 to TDES_MR.PKRS.

56.5.10 TDES Input Data Register x

Name: TDES_IDATARx
Offset: 0x40 + x*0x04 [x=0..1]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data
The two 32-bit TDES_IDATARx are used to set the 64-bit data block used for encryption/decryption.
TDES_IDATAR0.IDATA refers to the first word of the data to be encrypted/decrypted, and TDES_IDATAR1.IDATA to the last one.
These registers are write-only to prevent the input data from being read by another application.

56.5.11 TDES Output Data Register x

Name: TDES_ODATARx
Offset: 0x50 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The two 32-bit TDES_ODATARx contain the 64-bit data block which has been encrypted/decrypted. TDES_ODATAR0.ODATA refers to the first word, TDES_ODATAR1.ODATA to the last one.

56.5.12 TDES Initialization Vector Register x

Name: TDES_IVRx
Offset: 0x60 + x*0x04 [x=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

These registers are write-only to prevent the Initialization Vector from being read by another application.

These registers are not used for the ECB mode and must not be written.

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector
The two 32-bit TDES_IVRx are used to set the 64-bit initialization vector data block, which is used by some modes of operation as an additional initial input.
TDES_IVR1.IV refers to the first word of the Initialization Vector, TDES_IVR2.IV to the last one.

56.5.13 TDES XTEA Rounds Register

Name: TDES_XTEA_RNDR
Offset: 0x70
Reset: 0x000000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			XTEA_RNDS[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – XTEA_RNDS[5:0] Number of Rounds
This 6-bit field is used to define the number of complete rounds (1 complete round = 2 Feistel rounds) processed in XTEA algorithm.
The value of XTEA_RNDS has no effect if TDES_MR.TDESMOD is set to 0x0 or 0x1.
0x00 corresponds to 1 complete round, 0x01 corresponds to 2 complete rounds, etc.

56.5.14 TDES Write Protection Mode Register

Name: TDES_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x444553	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags are set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the TDES_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.

Value	Name	Description
5	CLEAR_CGD_SEQE	If a processing is in progress when the TDES_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TDES_WPSR.WPVSRC and the last software control error type is reported in TDES_WPSR.SWETYP. The TDES_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TDES_WPSR.WPVSRC and only the first software control error type is reported in TDES_WPSR.SWETYP. The TDES_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to0x444553 (“DES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x444553 (“DES” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x444553 (“DES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x444553 (“DES” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x444553 (“DES” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x444553 (“DES” in ASCII).

56.5.15 TDES Write Protection Status Register

Name: TDES_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality.
1	ERROR	An access is performed into key, input data, control registers while the TDES is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	TDES is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of TDES_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A key write, init value write, output data read, Mode register write, Private Key bus access or XTEA round register has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the keys are not fully loaded into TDES_KEYxWRy.

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Register Protection Violation Status (cleared on read)

Value	Description
0	No Private Key internal register access violation has occurred since the last read of TDES_WPSR.
1	A Private Key internal register access violation has occurred since the last read of TDES_WPSR.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TDES_WPSR.
1	A software error has occurred since the last read of TDES_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TDES_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TDES_WPSR. This flag is set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of TDES_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of TDES_WPSR. This flag is set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TDES_WPSR.
1	A write protection violation has occurred since the last read of TDES_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

57. True Random Number Generator (TRNG)

57.1 Description

The True Random Number Generator (TRNG) passes the American *NIST Special Publication 800-22 (A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications)* and the *Diehard Suite of Tests*.

The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

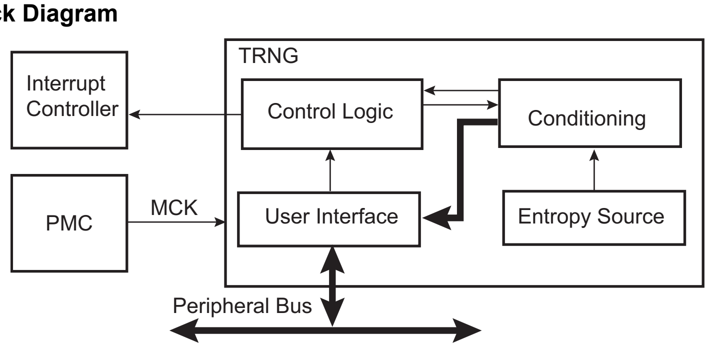
The TRNG is fully designed with digital cells, and under the specified operating conditions, external factors such as temperature, humidity, etc. affect TRNG ageing in the same manner as all other digital peripherals (CPU core, bus matrix, etc.) of the product.

57.2 Embedded Characteristics

- Passes *NIST Special Publication 800-22 Test Suite*
- Passes *Diehard Suite of Tests*
- Usable as Entropy Source for Seeding a NIST-approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit Random Number at Maximum 84 Clock Cycles
- Functional Safety Monitors and Reports:
  – Monitoring of internal sequencer abnormal states
  – Abnormal software access reports
  – Register write protection
- Private Key Bus Interface to Transfer Cryptographic Keys Not Readable From Any Peripheral Nor From Software

57.3 Block Diagram

Figure 57-1. TRNG Block Diagram



57.4 Product Dependencies

57.4.1 Power Management

The TRNG interface may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the TRNG user interface clock. The user interface clock is independent from any clock that may be used in the entropy source logic circuitry. The source of entropy can be enabled before enabling the user interface clock.

57.4.2 Interrupt Sources

The TRNG interface has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TRNG.

57.5 Functional Description

As soon as the TRNG is enabled in the Control register (TRNG_CR), the generator provides one 32-bit random value at a maximum streaming rate of 84 clock cycles. Entropy rate increases at a lower frequency. It is possible to divide by 2 the streaming rate by configuring the Mode register (TRNG_MR) to achieve better entropy if the streaming rate does not require new data every 84 clock cycles. For a lower streaming rate, the software intervention is required to skip, on a regular basis, the data ready information reported in the Status register (TRNG_ISR).

A sequence of random values can be generated by the TRNG and a random value can be directly loaded through the private key bus into specific private key internal registers of the private key bus clients (for example, AES or other encryption unit). There is no possibility of reading these keys from the processor and software from system bus. This is done by writing the Private Key Bus Control register (TRNG_PKBCR) with the appropriate destination encryption unit (KSLAVE), length of the key to be generated (KLENGTH) and TrustZone security attribute (KID). KID must correspond to the security level programmed in the MATRIX Security Peripheral Select x register for the destination encryption unit.

This random value transferred through the private key bus cannot be used for encrypted communications with remote equipment, but is useful while the system remains in Active mode to reinforce the security of data processed by the application running on the system and stored temporarily in external memories. The cryptography keys are never known to application software, thus they cannot be exchanged or provided to the external world in any case.

Note: Putting the system into Backup mode causes the key stored in the encryption engine to be lost. Local encryption and decryption of data can still be performed by using a key unknown to the software if the key is stored in a non-volatile area, for example in the OTP memory. The TRNG can transfer a random value to the non-volatile memory of the system to store the key for further decryption. When the key is required for decryption, it can be transferred by the private key bus from the OTP memory to the encryption module.

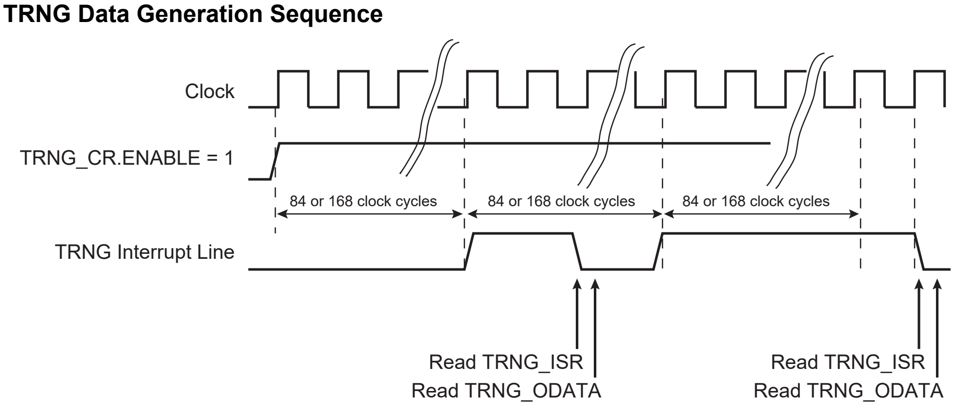
By writing a ‘1’ to the HALFR bit in the Mode register (TRNG_MR), the random values are provided every 168 cycles instead of every 84 cycles. HALFR must be written to ‘1’ when the TRNG peripheral clock frequency is above 100 MHz.

The TRNG interrupt line can be enabled in the Interrupt Enable register (TRNG_IER), and disabled in the Interrupt Disable register (TRNG_IDR). This interrupt is set when a new random value is available or when a transfer over the private key bus is complete and is cleared when the Status register (TRNG_ISR) is read. The flag TRNG_ISR.DATRDY is set when the random data is ready to be read out on the 32-bit Output Data register (TRNG_ODATA). The flag TRNG_ISR.EOTPKB is set when the transfer through the private key bus is complete.

Normal Operating Mode

The normal operating mode checks that the TRNG_ISR.DATRDY flag equals ‘1’ before reading TRNG_ODATA when a 32-bit random value is required by the software application.

Figure 57-2. TRNG Data Generation Sequence



Key Bus Operating Mode

After a write to KSLAVE, KID and KLENGTH in TRNG_PKBCR, the software:

- waits for the end of transfer of the key indicated by the TRNG_ISR.EOTPKB flag being read at '1', optionally after a TRNG interrupt,
- checks for any key bus access violation in the selected private key bus destination client status register,
- uses the private key bus destination client or launches any other private key bus transfer.

Figure 57-3. TRNG Private Key Bus

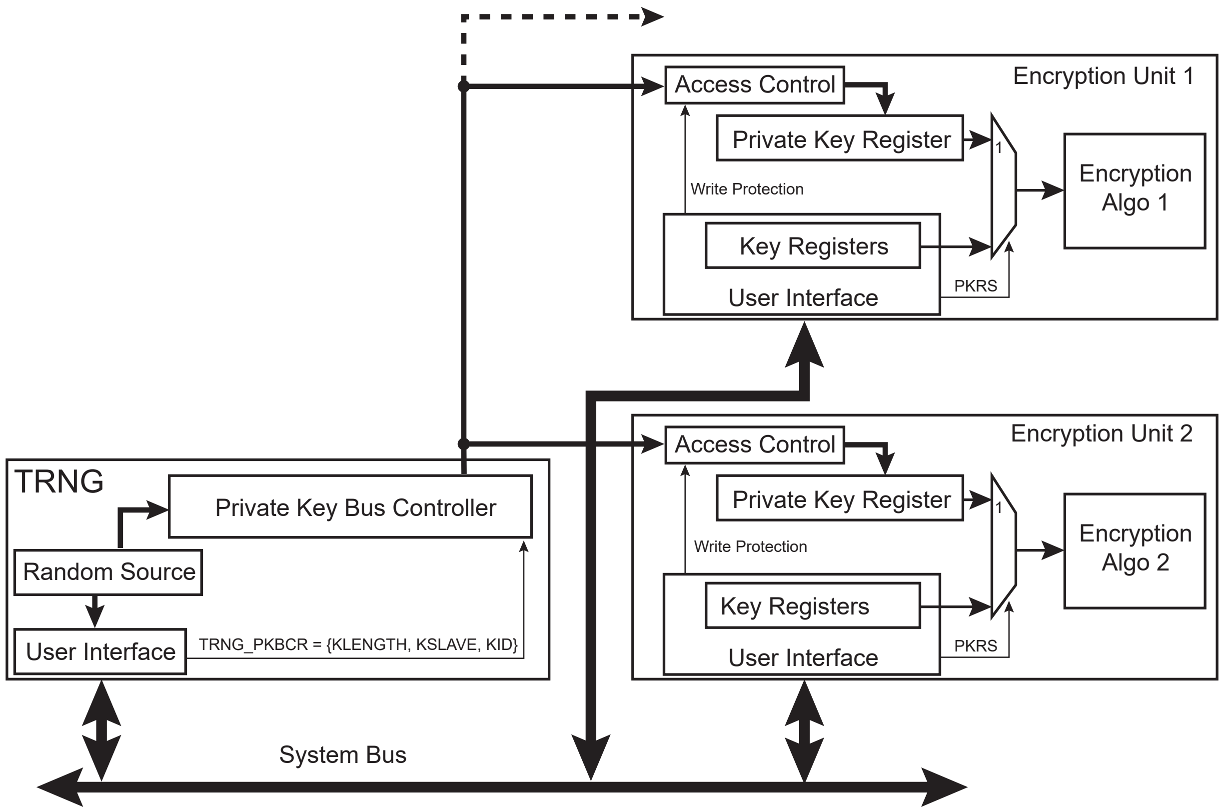
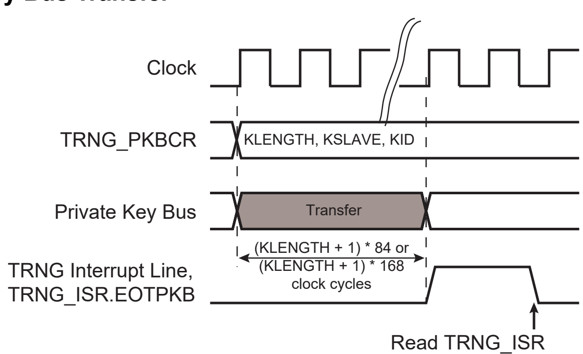


Figure 57-4. TRNG Private Key Bus Transfer



57.5.1 First Value Read after Power-up

After a power-up and the first configuration to enable the TRNG, the first data can be read as soon as the flag DATRDY is set in the Interrupt Status register (TRNG_ISR). However, randomness (entropy) of a sequence of first value read after a power-up sequence is correct only if the TRNG has been enabled for a significant period of time.

When the first value after power-up is a key factor for the application, it is recommended to wait for 5 ms before reading the first value and after the power-up followed by the initial enable of the TRNG.

57.5.2 Entropy

The TRNG provides a new random data at a maximum rate of peripheral clock divided by 84. However, entropy increases as the reading rate decreases.

57.5.3 Register Write Protection

To prevent any single software error from corrupting TRNG behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TRNG Write Protection Mode Register (TRNG_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TRNG Write Protection Status Register (TRNG_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically cleared by reading TRNG_WPSR.

The following register can be write-protected when WPEN is set:

- TRNG Mode Register

The following registers can be write-protected when WPITEN is set:

- TRNG Interrupt Enable Register
- TRNG Interrupt Disable Register

The following registers can be write-protected when WPCREN is set:

- TRNG Control Register
- TRNG Private Key Bus Control Register

57.5.4 Security and Functional Analysis and Reports

Several type of checks are performed when the TRNG is enabled.

The peripheral clock of the TRNG is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TRNG. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TRNG_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TRNG is also monitored and if an abnormal state is detected, the flag TRNG_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TRNG are monitored and if an incorrect access is performed, the flag TRNG_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the TRNG_WPSR.SWETYP field (see TRNG Write Protection Status Register for details). e.g., reading the TRNG_ODATA when the TRNG is disabled is an error, as well as reading the TRNG_ODATA, when the TRNG_ISR.DATRDY flag is cleared. TRNG_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TRNG_WPSR is read.

If one of these flags is set, the flag TRNG_ISR.SECE is set and can trigger an interrupt if the TRNG_IMR.SECE bit is ‘1’. SECE is cleared by reading TRNG_ISR.

57.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TRNG_CR	31:24	WAKEY[23:16]							
		23:16	WAKEY[15:8]							
		15:8	WAKEY[7:0]							
		7:0								ENABLE
0x04	TRNG_MR	31:24								
		23:16								
		15:8								
		7:0								HALFR
0x08	TRNG_PKBCR	31:24	WAKEY[15:8]							
		23:16	WAKEY[7:0]							
		15:8	KLENGTH[7:0]							
		7:0			KSLAVE[1:0]					KID
0x0C ... 0x0F	Reserved									
0x10	TRNG_IER	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x14	TRNG_IDR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x18	TRNG_IMR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x1C	TRNG_ISR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x20 ... 0x4F	Reserved									
0x50	TRNG_ODATA	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54 ... 0xE3	Reserved									
0xE4	TRNG_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	TRNG_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

57.6.1 TRNG Control Register

Name: TRNG_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								W
Reset								–

Bits 31:8 – WAKEY[23:0] Register Write Access Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation.

Bit 0 – ENABLE Enable TRNG to Provide Random Values

Value	Description
0	Disables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.
1	Enables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.

57.6.2 TRNG Mode Register

Name: TRNG_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								HALFR
Access								R/W
Reset								0

Bit 0 – HALFR Half Rate Enable

Value	Name	Description
0	DISABLED	Maximum stream rate provided (1 sample every 84 MCK clock cycles).
1	ENABLED	Half maximum stream rate provided if the peripheral clock frequency is above 100 MHz (1 sample every 168 MCK clock cycles).

57.6.3 TRNG Private Key Bus Control Register

Name: TRNG_PKBCR
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KLENGTH[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
			KSLAVE[1:0]					KID
Access			W	W				W
Reset			–	–				–

Bits 31:16 – WAKEY[15:0] Register Write Access Key

Value	Name	Description
0x524B	PASSWD	Writing any other value in this field aborts the write operation.

Bits 15:8 – KLENGTH[7:0] Key Length
Length-1 in 32-bit words of the key(s) to be directly loaded from the TRNG into the private key internal registers of the private key bus client KSLAVE.
Example: for one 64-bit key to be loaded, KLENGTH must be written to 1. For 128-bit keys, KLENGTH must be written to 3.

Bits 5:4 – KSLAVE[1:0] Key Bus Client
Private key bus client identifier for the destination encryption unit to be loaded from the TRNG.

Value	Name	Description
0	TDES_ID	TDES
1	AES_ID	AES
2	TZAESB_ID	TZAESB
3	OTPC_ID	OTPC

Bit 0 – KID Key ID
Private Key ID of the targeted private key bus client KSLAVE to be loaded from the TRNG.
TrustZone access protection is only done at the selected key bus client according to its TrustZone configuration.

Value	Name	Description
0	SECURE_KEY	TrustZone Secure Key access
1	NOT_SECURE_KEY	TrustZone Not Secure Key access

57.6.4 TRNG Interrupt Enable Register

Name: TRNG_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Enable

Bit 1 – SECE Security and/or Safety Event Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

57.6.5 TRNG Interrupt Disable Register

Name: TRNG_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Disable

Bit 1 – SECE Security and/or Safety Event Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

57.6.6 TRNG Interrupt Mask Register

Name: TRNG_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Mask

Bit 1 – SECE Security and/or Safety Event Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

57.6.7 TRNG Interrupt Status Register

Name: TRNG_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus (cleared on read)

Value	Description
0	No private key bus transfer has ended since the last read of the Interrupt Status Register.
1	The private key bus transfer has ended.

Bit 1 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No safety or security event occurred since the last read of the Interrupt Status Register.
1	One or more safety or security event occurred since the last read of TRNG_ISR. For details on the event, see TRNG Write Protection Status Register.

Bit 0 – DATRDY Data Ready (cleared on read)

Value	Description
0	Output data is not valid or TRNG is disabled.
1	New random value has been completed since the last read of TRNG_ISR.

57.6.8 TRNG Output Data Register

Name: TRNG_ODATA
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The 32-bit Output Data register contains the 32-bit random data.

57.6.9 TRNG Write Protection Mode Register

Name: TRNG_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TRNG_WPSR.WPVSRC and the last software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TRNG_WPSR.WPVSRC and only the first software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

57.6.10 TRNG Write Protection Status Register

Name: TRNG_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality.
1	ERROR	Reading TRNG_ODATA when TRNG is disabled or used for private key bus transfer does not provide a random value. Writing to the PKB_CTRL register while a private key bus transfer is ongoing does not launch a new private key bus transfer.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TRNG is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TRNG is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address.
3	TRNG_DIS	The TRNG_ODATA register has been read when TRNG is disabled or used for private key bus transfer (Error).
4	PKB_BUSY	A write access to the PKB_CTRL register has been attempted during a private key bus transfer (Error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TRNG_WPSR.

Value	Description
1	A software error has occurred since the last read of TRNG_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TRNG_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TRNG_WPSR. This flag is set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of TRNG_WPSR. Under normal operating conditions, this bit is always cleared.
1	A clock glitch has occurred since the last read of TRNG_WPSR. This flag is set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

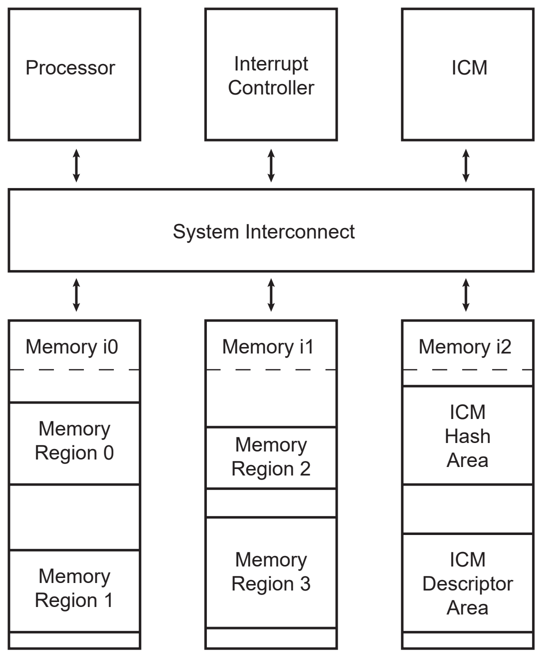
Value	Description
0	No write protection violation has occurred since the last read of TRNG_WPSR.
1	A write protection violation has occurred since the last read of TRNG_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

58. Integrity Check Monitor (ICM)

58.1 Description

The Integrity Check Monitor (ICM) is a DMA controller that performs hash calculation over multiple memory regions through the use of transfer descriptors located in memory (ICM Descriptor Area). The Hash function is based on the Secure Hash Algorithm (SHA). The ICM integrates two modes of operation. The first one is used to hash a list of memory regions and save the digests to memory (ICM Hash Area). The second mode is an active monitoring of the memory. In that mode, the hash function is evaluated and compared to the digest located at a predefined memory address (ICM Hash Area). If a mismatch occurs, an interrupt is raised. See the figure below for an example of four-region monitoring. Hash and Descriptor areas are located in Memory instance i2, and the four regions are split in memory instances i0 and i1.

Figure 58-1. Four-region Monitoring Example



The ICM SHA engine is compliant with the American FIPS (Federal Information Processing Standard) Publication 180-2 specification.

The following terms are concise definitions of the ICM concepts used throughout this document:

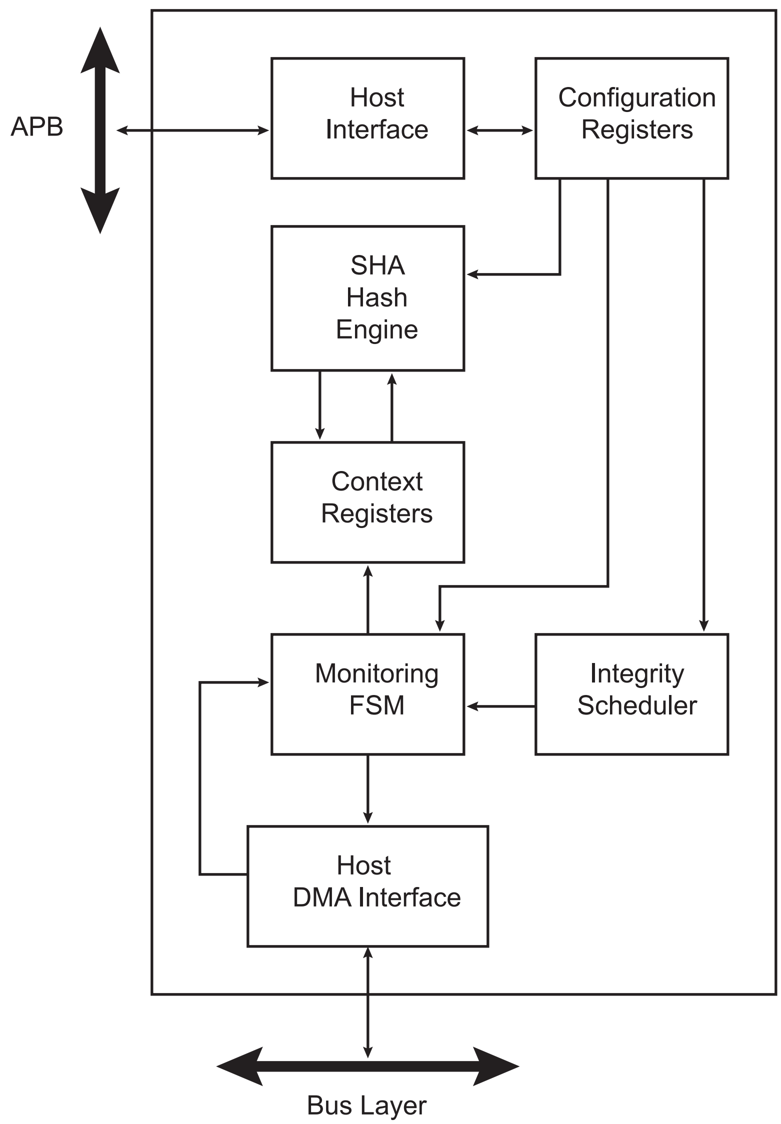
- Region—a partition of instruction or data memory space
- Region Descriptor—a data structure stored in memory, defining region attributes
- Region Attributes—region start address, region size, region SHA engine processing mode, Write Back or Compare function mode
- Context Registers—a set of ICM non-memory-mapped, internal registers which are automatically loaded, containing the attributes of the region being processed
- Main List—a list of region descriptors. Each element associates the start address of a region with a set of attributes.
- Secondary List—a linked list defined on a per region basis that describes the memory layout of the region (when the region is non-contiguous)
- Hash Area—predefined memory space where the region hash results (digest) are stored

58.2 Embedded Characteristics

- Host DMA Interface
- Supports Monitoring of up to 4 Non-Contiguous Memory Regions
- Supports Block Gathering Using Linked Lists
- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256)
- Compliant with FIPS Publication 180-2
- Configurable Processing Period:
  – When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
  – When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.
- Programmable Bus Burden
- Register Write Protection

58.3 Block Diagram

Figure 58-2. ICM Block Diagram



58.4 Product Dependencies

58.4.1 Power Management

The peripheral clock is not continuously provided to the ICM. The programmer must first enable the ICM clock in the Power Management Controller (PMC) before using the ICM.

58.4.2 Interrupt Sources

The ICM has an interrupt lineconnected to the interrupt controller. Handling the ICM interrupt requires programming the interrupt controller before configuring the ICM.

58.5 Functional Description

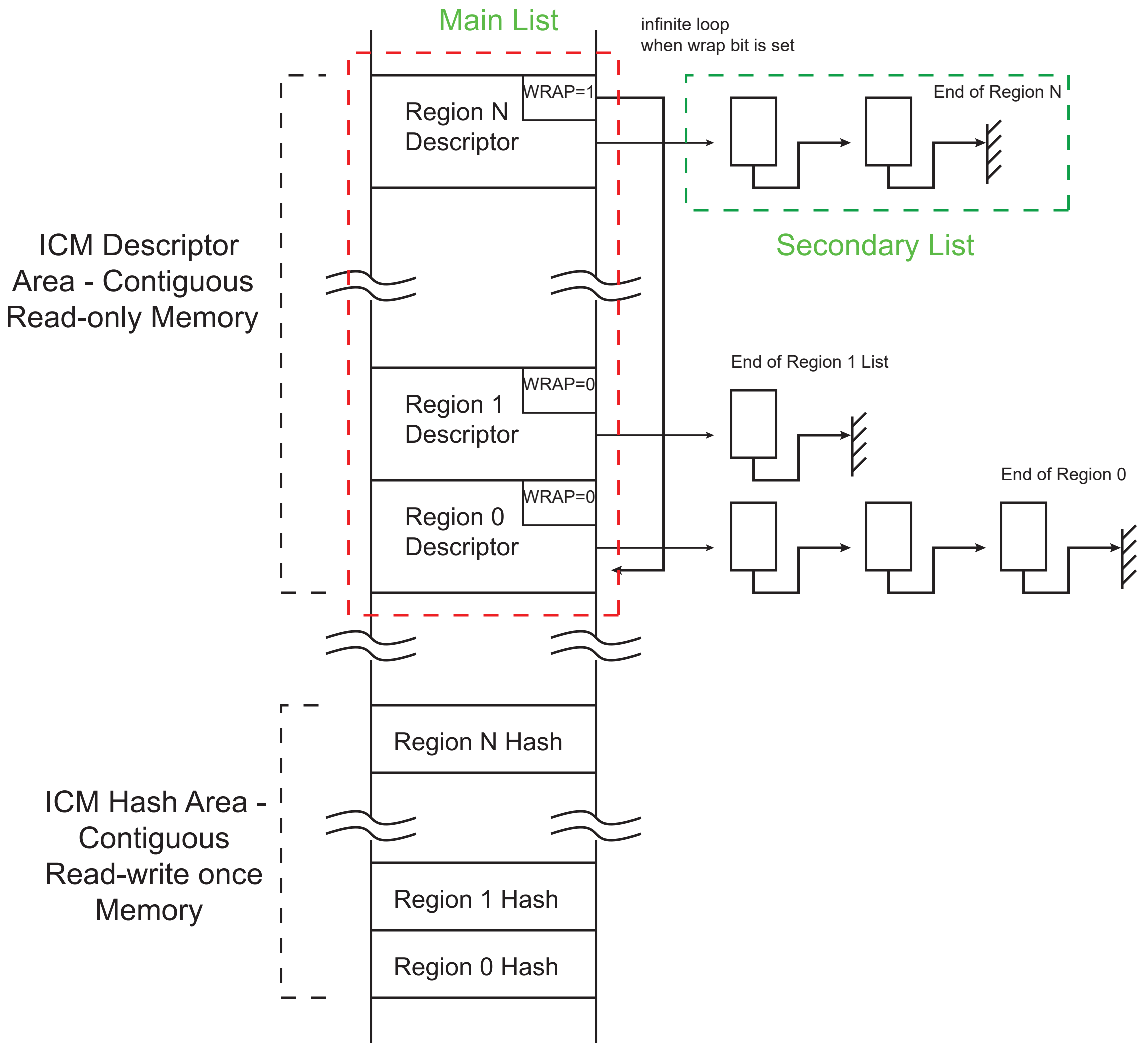
58.5.1 Overview

The Integrity Check Monitor (ICM) is a DMA controller that performs SHA-based memory hashing over memory regions. As shown in figure Integrity Check Monitor Block Diagram, it integrates a DMA interface, a Monitoring Finite State Machine (FSM), an integrity scheduler, a set of context registers, a SHA engine, an interface for configuration and status registers.

The ICM integrates a Secure Hash Algorithm engine (SHA). This engine requires a message padded according to FIPS180-2 specification when used as a SHA calculation unit only. Otherwise, if the ICM is used as integrated check for memory content, the padding is not mandatory. The SHA module produces an N-bit message digest each time a block is read and a processing period ends. N is 160 for SHA1, 224 for SHA224, 256 for SHA256.

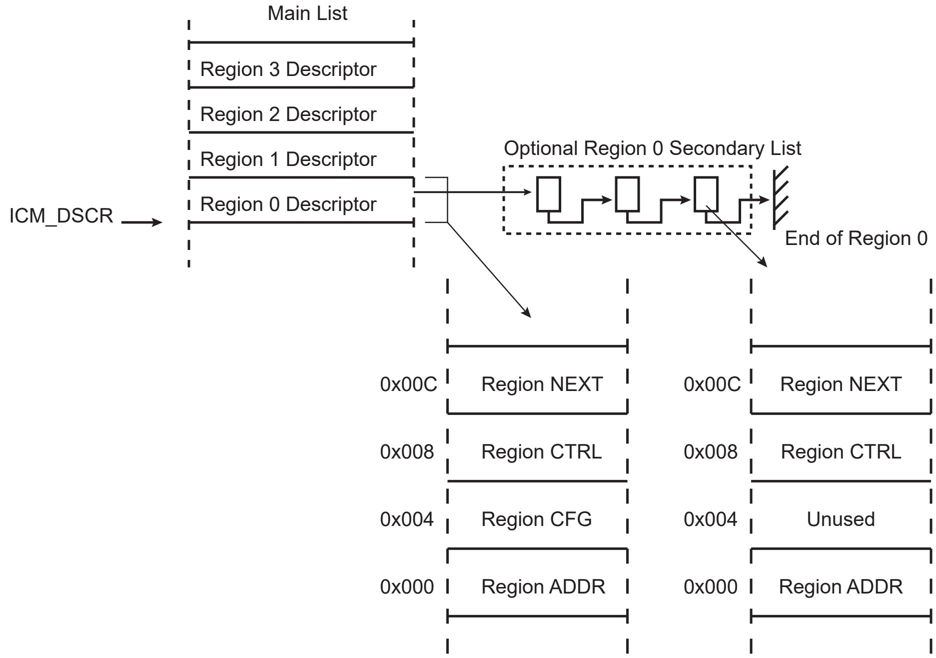
When the ICM module is enabled, it sequentially retrieves a circular list of region descriptors from the memory (Main List described in figure ICM Region Descriptor and Hash Areas). Up to four regions may be monitored. Each region descriptor is composed of four words indicating the layout of the memory region (see figure Region Descriptor). It also contains the hashing engine configuration on a per-region basis. As soon as the descriptor is loaded from the memory and context registers are updated with the data structure, the hashing operation starts. A programmable number of blocks (see TRSIZE field of the ICM_RCTRL structure member) is transferred from the memory to the SHA engine. When the desired number of blocks have been transferred, the digest is either moved to memory (Write Back function) or compared with a digest reference located in the system memory (Compare function). If a digest mismatch occurs, an interrupt is triggered if unmasked. The ICM module passes through the region descriptor list until the end of the list marked by an end of list marker (WRAP or EOM bit in ICM_RCFG structure member set to one). To continuously monitor the list of regions, the WRAP bit must be set to one in the last data structure and EOM must be cleared.

Figure 58-3. ICM Region Descriptor and Hash Areas



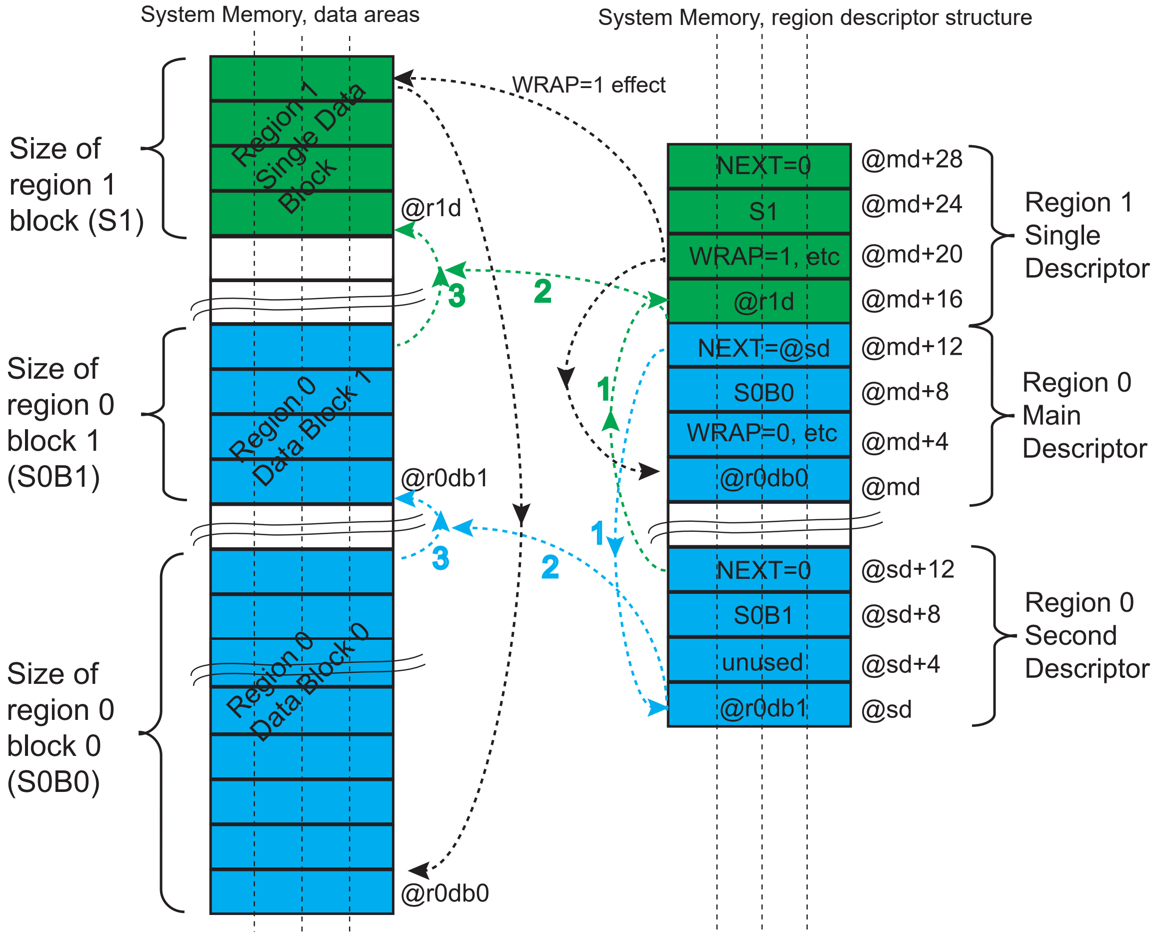
Each region descriptor supports gathering of data through the use of the Secondary List. Unlike the Main List, the Secondary List cannot modify the configuration attributes of the region. When the end of the Secondary List has been encountered, the ICM returns to the Main List. Memory integrity monitoring can be considered as a background service and the mandatory bandwidth shall be very limited. In order to limit the ICM memory bandwidth, use ICM_CFG.BBC to control the ICM memory load.

Figure 58-4. Region Descriptor



The figure below shows an example of the mandatory ICM settings required to monitor three memory data blocks of the system memory (defined as two regions) with one region being not contiguous (two separate areas) and one contiguous memory area. For each region, the SHA algorithm may be independently selected (different for each region). The wrap allows continuous monitoring.

Figure 58-5. Example: Monitoring of 3 Memory Data Blocks (Defined as 2 Regions)



58.5.2 ICM Region Descriptor Structure

The ICM Region Descriptor Area is a contiguous area of system memory that the controller and the processor can access. When the ICM is activated, the controller performs a descriptor fetch operation at *(ICM_DSCR) address. If the Main List contains more than one descriptor (i.e., more than one region is to be monitored), the fetch address is *(ICM_DSCR) + (RID<<4) where RID is the region identifier.

Table 58-1. Region Descriptor Structure (Main List)

Offset	Structure Member	Name
ICM_DSCR+0x000+RID*(0x10)	ICM Region Start Address	ICM_RADDR
ICM_DSCR+0x004+RID*(0x10)	ICM Region Configuration	ICM_RCFG
ICM_DSCR+0x008+RID*(0x10)	ICM Region Control	ICM_RCTRL
ICM_DSCR+0x00C+RID*(0x10)	ICM Region Next Address	ICM_RNEXT

58.5.2.1 ICM Region Start Address Structure Member

Name: ICM_RADDR
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x000+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
	RADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	23	22	21	20	19	18	17	16
	RADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	15	14	13	12	11	10	9	8
	RADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	7	6	5	4	3	2	1	0
	RADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 31:0 – RADDR[31:0] Region Start Address
This field indicates the first byte address of the region.

58.5.2.2 ICM Region Configuration Structure Member

Name: ICM_RCFG
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x004+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bits 14:12 – ALGO[2:0] SHA Algorithm
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Bit 10 – PROCDLY Processing Delay
When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one.
1	LONGEST	SHA processing runtime is the longest one.

Bit 9 – SUIEN Monitoring Status Updated Condition Interrupt (Default Enabled)

Value	Description
0	The ICM_ISR.RSU[i] flag is set when the corresponding descriptor is loaded from memory to ICM.
1	The ICM_ISR.RSU[i] flag remains cleared even if the setting condition is met.

Bit 8 – ECIEN End Bit Condition Interrupt (Default Enabled)

Value	Description
0	The ICM_ISR.REC[i] flag is set when the descriptor with the EOM bit set is processed.
1	The ICM_ISR.REC[i] flag remains cleared even if the setting condition is met.

Bit 7 – WCIEN Wrap Condition Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RWC[i] flag is set when the WRAP bit is set in a descriptor of the main list.
1	ICM_ISR.RWC[i] flag remains cleared even if the setting condition is met.

Bit 6 – BEIEN Bus Error Interrupt Disable (Default Enabled)

Value	Description
0	The flag is set when an error is reported on the system bus by the bus matrix.
1	The flag remains cleared even if the setting condition is met.

Bit 5 – DMIEN Digest Mismatch Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RBE[i] flag is set when the hash value just calculated from the processed region differs from expected hash value.
1	The ICM_ISR.RBE[i] flag remains cleared even if the setting condition is met.

Bit 4 – RHIEN Region Hash Completed Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RHC[i] flag is set when the field NEXT = 0 in a descriptor of the main or second list.
1	The ICM_ISR.RHC[i] flag remains cleared even if the setting condition is met.

Bit 2 – EOM End Of Monitoring

Value	Description
0	The current descriptor does not terminate the monitoring.
1	The current descriptor terminates the Main List. WRAP value has no effect.

Bit 1 – WRAP Wrap Command

Value	Description
0	The next region descriptor address loaded is the current region identifier descriptor address incremented by 0x10.
1	The next region descriptor address loaded is ICM_DSCR.

Bit 0 – CDWBN Compare Digest or Write Back Digest

Value	Description
0	The digest is written to the Hash area.
1	The digest value is compared to the digest stored in the Hash area.

58.5.2.3 ICM Region Control Structure Member

Name: ICM_RCTRL
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x008+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TRSIZE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TRSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 15:0 – TRSIZE[15:0] Transfer Size for the Current Chunk of Data

58.5.2.4 ICM Region Next Address Structure Member

Name: ICM_RNEXT
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x00C+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
	NEXT[28:21]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	NEXT[20:13]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	NEXT[12:5]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	NEXT[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset								

Bits 31:3 – NEXT[28:0] Region Transfer Descriptor Next Address
When configured to 0, this field indicates that the current descriptor is the last descriptor of the Secondary List, otherwise it points at a new descriptor of the Secondary List.

58.5.3 ICM Hash Area

The ICM Hash Area is a contiguous area of system memory that the controller and the processor can access. The physical location is configured in the ICM hash area start address register. This address is a multiple of 128 bytes. If the CDWBN bit of the context register is cleared (i.e., Write Back activated), the ICM performs a digest write operation at the following starting location: *(ICM_HASH) + (RID<<5), where RID is the current region context identifier. If the CDWBN bit of the context register is set (i.e., Digest Comparison activated), the ICM performs a digest read operation at the same address.

58.5.3.1 Message Digest Example

Considering the following 512-bit message (example given in FIPS 180-2):

“61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018”

The message is written to memory in a Little Endian (LE) system architecture.

Table 58-2. 512-bit Message Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	00	00	00	00	80	63	62	61
0x008–0x030	00	00	00	00	00	00	00	00
0x038	18	00	00	00	00	00	00	00

The digest is stored at the memory location pointed at by the ICM_HASH pointer with a Region Offset.

Table 58-3. LE Resulting SHA-160 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	6a	81	06	47	36	3e	99	a9
0x008	6c	c2	50	78	71	25	3e	ba
0x010	–	–	–	–	9d	d8	d0	9c

Table 58-4. Resulting SHA-256 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	ea	cf	01	8f	bf	16	78	ba
0x008	23	22	ae	5d	de	40	41	41
0x010	9c	7a	17	96	a3	61	03	b0
0x018	ad	15	00	f2	61	ff	10	b4

Considering the following 1024-bit message (example given in FIPS 180-2):

“61626380000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000018”

The message is written to memory in a Little Endian (LE) system architecture.

Table 58-5. 1024 bits Message Memory Mapping

Memory Address	Address Offset / Byte Lane							
	0x7 / 63:56	0x6 / 55:48	0x5 / 47:40	0x4 / 39:32	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	00	00	00	00	80	63	62	61
0x008–0x070	00	00	00	00	00	00	00	00
0x078	18	00	00	00	00	00	00	00

58.5.4 Using ICM as SHA Engine

The ICM can be configured to only calculate a SHA1, SHA224, SHA256 digest value.

58.5.4.1 Settings for Simple SHA Calculation

The start address of the system memory containing the data to hash must be configured in the transfer descriptor of the DMA embedded in the ICM.

The transfer descriptor is a system memory area integer multiple of 4 x 32-bit words and the start address of the descriptor must be configured in ICM_DSCR (the start address must be aligned on 64-bytes; six LSB must be cleared). If the data to hash is already padded according to SHA standards, only a single descriptor is required, and ICM_RCFG.EOM must be written to 1. If the data to hash does not contain a padding area, it is possible to define the padding area in another system memory location, the ICM can be configured to automatically jump from a memory area to another one by configuring the descriptor register ICM_RNEXT with a value that differs from 0. Configuring ICM_RNEXT.NEXT with the start address of the padding area forces the ICM to concatenate both areas, thus providing the SHA result from the start address of the hash area configured in ICM_HASH.

Whether the system memory is configured as a single or multiple data block area, ICM_RCFG.CDWBN and ICM_RCFG.WRAP must be cleared. The bits WBDIS, EOMDIS, SLBDIS must be cleared in ICM_CFG.

ICM_RCTRL.RHIEN and ICM_RCTRL.ECIEN must be written to 1. The flag RHC[i], i being the region index, is set (if RHIEN is set) when the hash result is available at address defined in ICM_HASH. The flag REC[i], i being the region index, is set (if ECIEN is set) when the hash result is available at the address defined in ICM_HASH.

An interrupt is generated if the bit RHC[i] is written to 1 in the ICM_IER (if RHC[i] is set in ICM_RCTRL of region i) or if the bit REC[i] is written to 1 in the ICM_IER (if REC[i] is set in ICM_RCTRL of region i).

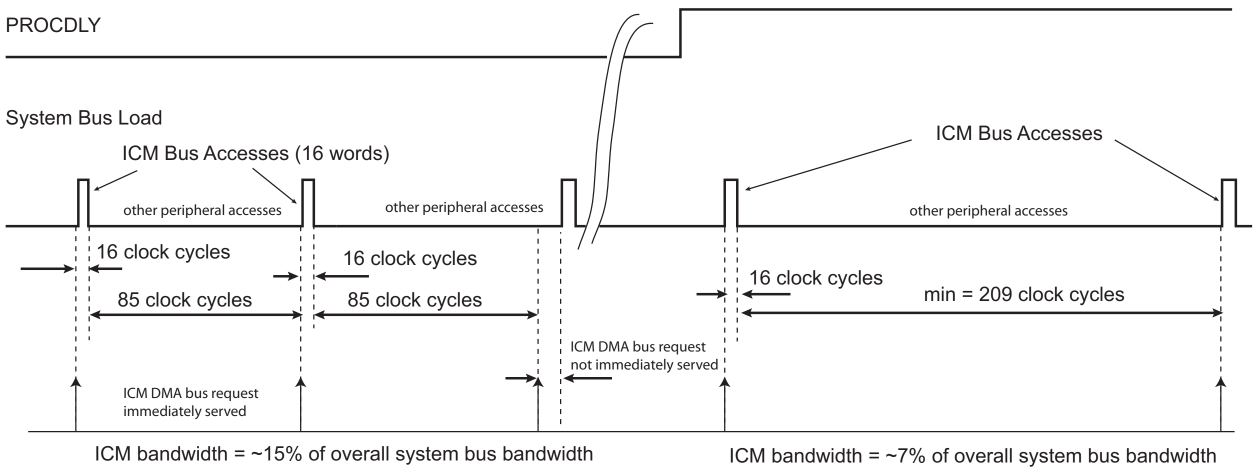
58.5.4.2 Processing Period

The ICM engine has a core (SHA) inherent processing period that may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, DMA). The ICM SHA engine processing period can be configured to reduce the bandwidth required by writing ICM_RCFG.PROCDLY=1.

In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization (ICM_RCFG.PROCDLY=0). The longest period is 209 clock cycles + 2 clock cycles when ICM_RCFG.PROCDLY=1 (see the figure below).

In SHA256 or SHA224 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization. The longest period is 194 clock cycles + 2 clock cycles.

Figure 58-6. Bandwidth Usage in SHA1 Mode



58.5.5 ICM Automatic Monitoring Mode

ICM_CFG.ASCD is used to activate the ICM Automatic Monitoring mode. When ICM_CFG.ASCD is set and bits CDWBN and EOM in ICM.RCFG equal 0, the ICM performs the following actions:

1. The ICM passes through the Main List once to calculate the message digest of the monitored area.
2. When WRAP = 1 in ICM_RCFG, the ICM begins monitoring. CDWBN in ICM_RCFG is now automatically set and EOM is cleared. These bits have no effect during the monitoring period that ends when EOM is set.

58.5.6 Programming the ICM

Table 58-6. Region Attributes

Transfer Type		Main List	ICM_RCFG			ICM_RNEXT	Comments
			CDWBN	WRAP	EOM	NEXT	
Single Region	Contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	0	The Main List contains only one descriptor. The Secondary List is empty for that descriptor. The digest is computed and saved to memory.
	Non-contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	Secondary List address of the current region identifier	The Main List contains only one descriptor. The Secondary List describes the layout of the non-contiguous region.
	Contiguous list of blocks Digest comparison enabled Monitoring enabled	1 item	1	1	0	0	When the hash computation is terminated, the digest is compared with the one saved in memory.
Multiple Regions	Contiguous list of blocks Digest written to memory Monitoring disabled	More than one item	0	0	1 for the last, 0 otherwise	0	ICM passes through the list once.
	Contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1 for the last, 0 otherwise	0	0	ICM performs active monitoring of the regions. If a mismatch occurs, an interrupt is raised.
	Non-contiguous list of blocks Digest is written to memory Monitoring is disabled	More than one item	0	0	1	Secondary List address	ICM performs hashing and saves digests to the Hash area.
	Non-contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1	0	Secondary List address	ICM performs data gathering on a per region basis.

58.5.7 Security Features

When an undefined register access occurs, the URAD bit in the Interrupt Status Register (ICM_ISR) is set if unmasked. Its source is then reported in the Undefined Access Status Register (ICM_UASR). Only the first undefined register access is available through the ICM_UASR.URAT field.

Several kinds of unspecified register accesses can occur:

- Unspecified structure member set to one detected when the descriptor is loaded
- Configuration register (ICM_CFG) modified during active monitoring
- Descriptor register (ICM_DSCR) modified during active monitoring
- Hash register (ICM_HASH) modified during active monitoring
- Write-only register read access

The URAD bit and the URAT field can only be reset by writing a 1 to the ICM_CTRL.SWRST bit.

58.5.8 ICM Register Write Protection

To prevent any single software error from corrupting ICM behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the ICM Write Protection Mode Register (ICM_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the ICM Write Protection Status Register (ICM_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading ICM_WPSR.

The following register(s) can be write-protected when WPEN is set:

- ICM Configuration Register
- ICM Descriptor Area Start Address Register
- ICM Hash Area Start Address Register
- ICM User Initial Hash Value Register

The following registers can be write-protected when WPITEN is set:

- ICM Interrupt Enable Register
- ICM Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- ICM Control Register

58.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ICM_CFG	31:24								
		23:16								
		15:8	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
		7:0	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
0x04	ICM_CTRL	31:24								
		23:16								
		15:8	RMEN[3:0]				RMDIS[3:0]			
		7:0	REHASH[3:0]					SWRST	DISABLE	ENABLE
0x08	ICM_SR	31:24								
		23:16								
		15:8	RMDIS[3:0]				RAWRMDIS[3:0]			
		7:0								ENABLE
0x0C ... 0x0F	Reserved									
0x10	ICM_IER	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x14	ICM_IDR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x18	ICM_IMR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x1C	ICM_ISR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x20	ICM_UASR	31:24								
		23:16								
		15:8								
		7:0						URAT[2:0]		
0x24 ... 0x2F	Reserved									
0x30	ICM_DSCR	31:24	DASA[25:18]							
		23:16	DASA[17:10]							
		15:8	DASA[9:2]							
		7:0	DASA[1:0]							
0x34	ICM_HASH	31:24	HASA[24:17]							
		23:16	HASA[16:9]							
		15:8	HASA[8:1]							
		7:0	HASA[0]							
0x38	ICM_UIHVAL0	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x3C	ICM_UIHVAL1	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	ICM_UIHVAL2	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x44	ICM_UIHVAL3	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x48	ICM_UIHVAL4	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x4C	ICM_UIHVAL5	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x50	ICM_UIHVAL6	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x54	ICM_UIHVAL7	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x58 ... 0xE3	Reserved									
0xE4	ICM_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	ICM_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS

58.6.1 ICM Configuration Register

Name: ICM_CFG
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 15:13 – UALGO[2:0] User SHA Algorithm

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Bit 12 – UIHASH User Initial Hash Value

Value	Description
0	The secure hash standard provides the initial hash value.
1	The initial hash value is programmable. Field UALGO provides the SHA algorithm. The ALGO field of the ICM_RCFG structure member has no effect.

Bit 9 – DUALBUFF Dual Input Buffer

Value	Description
0	Dual Input Buffer mode is disabled.
1	Dual Input Buffer mode is enabled (better performances, higher bandwidth required on system bus).

Bit 8 – ASCD Automatic Switch To Compare Digest

Value	Description
0	Automatic monitoring mode is disabled.
1	The ICM passes through the Main List once to calculate the message digest of the monitored area. When WRAP = 1 in ICM_RCFG, the ICM begins monitoring.

Bits 7:4 – BBC[3:0] Bus Burden Control
This field is used to control the burden of the ICM system bus. The number of system clock cycles between the end of the current processing and the next block transfer is set to 2^{BBC}. Up to 32,768 cycles can be inserted.

Bit 2 – SLBDIS Secondary List Branching Disable

Value	Description
0	Branching to the Secondary List is permitted.
1	Branching to the Secondary List is forbidden. The NEXT field of the ICM_RNEXT structure member has no effect and is always considered as zero.

Bit 1 – EOMDIS End of Monitoring Disable

Value	Description
0	End of Monitoring is permitted.
1	End of Monitoring is forbidden. The EOM bit of the ICM_RCFG structure member has no effect.

Bit 0 – WBDIS Write Back Disable
When ASCD is set, WBDIS has no effect.

Value	Description
0	Write Back operations are permitted.
1	Write Back operations are forbidden. Context register CDWBN bit is internally set to one and cannot be modified by a linked list element. ICM_RCFG.CDWBN has no effect.

58.6.2 ICM Control Register

Name: ICM_CTRL
Offset: 0x04
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RMEN[3:0]				RMDIS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	REHASH[3:0]					SWRST	DISABLE	ENABLE
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bits 15:12 – RMEN[3:0] Region Monitoring Enable
Monitoring is activated by default.

Value	Description
0	No effect
1	When bit RMEN[i] is set to one, the monitoring of region with identifier i is activated.

Bits 11:8 – RMDIS[3:0] Region Monitoring Disable

Value	Description
0	No effect
1	When bit RMDIS[i] is set to one, the monitoring of region with identifier i is disabled.

Bits 7:4 – REHASH[3:0] Recompute Internal Hash

Value	Description
0	No effect
1	When REHASH[i] is set to one, Region i digest is re-computed. This bit is only available when region monitoring is disabled.

Bit 2 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the ICM.

Bit 1 – DISABLE ICM Disable Register

Value	Description
0	No effect
1	The ICM is disabled. If a region is active, this region is terminated.

Bit 0 – ENABLE ICM Enable

Value	Description
0	No effect
1	When set to one, the ICM is activated.

58.6.3 ICM Status Register

Name: ICM_SR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RMDIS[3:0]				RAWRMDIS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R
Reset								0

Bits 15:12 – RMDIS[3:0] Region Monitoring Disabled Status

Value	Description
0	Region i is being monitored (occurs after integrity check value has been calculated and written to Hash area).
1	Region i monitoring is not being monitored.

Bits 11:8 – RAWRMDIS[3:0] Region Monitoring Disabled Raw Status

Value	Description
0	Region i monitoring has been activated by writing a 1 in RMEN[i] of ICM_CTRL.
1	Region i monitoring has been deactivated by writing a 1 in RMDIS[i] of ICM_CTRL.

Bit 0 – ENABLE ICM Enable Register

Value	Description
0	ICM is disabled.
1	ICM is activated.

58.6.4 ICM Interrupt Enable Register

Name: ICM_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								URAD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – URAD Undefined Register Access Detection Interrupt Enable

Value	Description
0	No effect.
1	The Undefined Register Access interrupt is enabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Disable

Value	Description
0	No effect.
1	When RSU[i] is set to one, the region i Status Updated interrupt is enabled.

Bits 19:16 – REC[3:0] Region End bit Condition Detected Interrupt Enable

Value	Description
0	No effect.
1	When REC[i] is set to one, the region i End bit Condition interrupt is enabled.

Bits 15:12 – RWC[3:0] Region Wrap Condition detected Interrupt Enable

Value	Description
0	No effect.
1	When RWC[i] is set to one, the Region i Wrap Condition interrupt is enabled.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Enable

Value	Description
0	No effect.
1	When RBE[i] is set to one, the Region i Bus Error interrupt is enabled.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Enable

Value	Description
0	No effect.
1	When RDM[i] is set to one, the Region i Digest Mismatch interrupt is enabled.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Enable

Value	Description
0	No effect.
1	When RHC[i] is set to one, the Region i Hash Completed interrupt is enabled.

58.6.5 ICM Interrupt Disable Register

Name: ICM_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								URAD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – URAD Undefined Register Access Detection Interrupt Disable

Value	Description
0	No effect.
1	Undefined Register Access Detection interrupt is disabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Disable

Value	Description
0	No effect.
1	When RSU[i] is set to one, the region i Status Updated interrupt is disabled.

Bits 19:16 – REC[3:0] Region End bit Condition detected Interrupt Disable

Value	Description
0	No effect.
1	When REC[i] is set to one, the region i End bit Condition interrupt is disabled.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected Interrupt Disable

Value	Description
0	No effect.
1	When RWC[i] is set to one, the Region i Wrap Condition interrupt is disabled.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Disable

Value	Description
0	No effect.
1	When RBE[i] is set to one, the Region i Bus Error interrupt is disabled.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Disable

Value	Description
0	No effect.
1	When RDM[i] is set to one, the Region i Digest Mismatch interrupt is disabled.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Disable

Value	Description
0	No effect.
1	When RHC[i] is set to one, the Region i Hash Completed interrupt is disabled.

58.6.6 ICM Interrupt Mask Register

Name: ICM_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								R
Reset								0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	RSU[3:0]				REC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	RWC[3:0]				RBE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RDM[3:0]				RHC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – URAD Undefined Register Access Detection Interrupt Mask

Value	Description
0	Interrupt is disabled
1	Interrupt is enabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Mask

Value	Description
0	When RSU[i] is set to zero, the interrupt is disabled for region i.
1	When RSU[i] is set to one, the interrupt is enabled for region i.

Bits 19:16 – REC[3:0] Region End Bit Condition Detected Interrupt Mask

Value	Description
0	When REC[i] is set to zero, the interrupt is disabled for region i.
1	When REC[i] is set to one, the interrupt is enabled for region i.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected Interrupt Mask

Value	Description
0	When RWC[i] is set to zero, the interrupt is disabled for region i.
1	When RWC[i] is set to one, the interrupt is enabled for region i.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Mask

Value	Description
0	When RBE[i] is set to zero, the interrupt is disabled for region i.
1	When RBE[i] is set to one, the interrupt is enabled for region i.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Mask

Value	Description
0	When RDM[i] is set to zero, the interrupt is disabled for region i.

Value	Description
1	When RDM[i] is set to one, the interrupt is enabled for region i.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Mask

Value	Description
0	When RHC[i] is set to zero, the interrupt is disabled for region i.
1	When RHC[i] is set to one, the interrupt is enabled for region i.

58.6.7 ICM Interrupt Status Register

Name: ICM_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – URAD Undefined Register Access Detection Status
The URAD bit is only reset by the SWRST bit in ICM_CTRL.
The URAT field in ICM_UASR indicates the unspecified access type.

Value	Description
0	No undefined register access has been detected since the last SWRST.
1	At least one undefined register access has been detected since the last SWRST.

Bits 23:20 – RSU[3:0] Region Status Updated Detected
When RSU[i] is set, it indicates that a region status updated condition has been detected.

Bits 19:16 – REC[3:0] Region End Bit Condition Detected
When REC[i] is set, it indicates that an end bit condition has been detected.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected
When RWC[i] is set, it indicates that a wrap condition has been detected.

Bits 11:8 – RBE[3:0] Region Bus Error
When RBE[i] is set, it indicates that a bus error has been detected while hashing memory region i.

Bits 7:4 – RDM[3:0] Region Digest Mismatch
When RDM[i] is set, it indicates that there is a digest comparison mismatch between the hash value of the region with identifier i and the reference value located in the Hash Area.

Bits 3:0 – RHC[3:0] Region Hash Completed
When RHC[i] is set, it indicates that the ICM has completed the region with identifier i.

58.6.8 ICM Undefined Access Status Register

Name: ICM_UASR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						URAT[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – URAT[2:0] Undefined Register Access Trace
Only the first Undefined Register Access Trace is available through the URAT field.
The URAT field is only reset by the SWRST bit in the ICM_CTRL register.

Value	Name	Description
0	UNSPEC_STRUCT_MEMBER	Unspecified structure member set to one detected when the descriptor is loaded.
1	ICM_CFG_MODIFIED	ICM_CFG modified during active monitoring.
2	ICM_DSCR_MODIFIED	ICM_DSCR modified during active monitoring.
3	ICM_HASH_MODIFIED	ICM_HASH modified during active monitoring
4	READ_ACCESS	Write-only register read access

58.6.9 ICM Descriptor Area Start Address Register

Name: ICM_DSCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DASA[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DASA[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DASA[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DASA[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bits 31:6 – DASA[25:0] Descriptor Area Start Address
The start address is a multiple of the total size of the data structure (64 bytes).

58.6.10 ICM Hash Area Start Address Register

Name: ICM_HASH
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	HASA[24:17]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HASA[16:9]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HASA[8:1]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HASA[0]							
Access	R/W							
Reset	0							

Bits 31:7 – HASA[24:0] Hash Area Start Address
This field points at the Hash memory location. The address must be a multiple of 128 bytes.

58.6.11 ICM User Initial Hash Value Register

Name: ICM_UIHVALx
Offset: 0x38 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the ICM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VAL[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	VAL[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	VAL[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	VAL[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – VAL[31:0] Initial Hash Value
When ICM_CFG.UIHASH is set, the Initial Hash Value is user-programmable.
To meet the desired standard, use the following example values.
For ICM_UIHVAL0 field:

Example	Comment
0x67452301	SHA1 algorithm
0xC1059ED8	SHA224 algorithm
0x6A09E667	SHA256 algorithm

For ICM_UIHVAL1 field:

Example	Comment
0xEFCDAB89	SHA1 algorithm
0x367CD507	SHA224 algorithm
0xBB67AE85	SHA256 algorithm

For ICM_UIHVAL2 field:

Example	Comment
0x98BADCFE	SHA1 algorithm
0x3070DD17	SHA224 algorithm
0x3C6EF372	SHA256 algorithm

For ICM_UIHVAL3 field:

Example	Comment
0x10325476	SHA1 algorithm

...........continued	
Example	**Comment**
0xF70E5939	SHA224 algorithm
0xA54FF53A	SHA256 algorithm

For ICM_UIHVAL4 field:

Example	Comment
0xC3D2E1F0	SHA1 algorithm
0xFFC00B31	SHA224 algorithm
0x510E527F	SHA256 algorithm

For ICM_UIHVAL5 field:

Example	Comment
0x68581511	SHA224 algorithm
0x9B05688C	SHA256 algorithm

For ICM_UIHVAL6 field:

Example	Comment
0x64F98FA7	SHA224 algorithm
0x1F83D9AB	SHA256 algorithm

For ICM_UIHVAL7 field:

Example	Comment
0xBEFA4FA4	SHA224 algorithm
0x5BE0CD19	SHA256 algorithm

Example of Initial Value for SHA-1 Algorithm

Register Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000 ICM_UIHVAL0	01	23	45	67
0x004 ICM_UIHVAL1	89	ab	cd	ef
0x008 ICM_UIHVAL2	fe	dc	ba	98
0x00C ICM_UIHVAL3	76	54	32	10
0x010 ICM_UIHVAL4	f0	e1	d2	c3

58.6.12 ICM Write Protection Mode Register

Name: ICM_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x49434D	PASSWD	Writing any other value in this field aborts the write operation of the WPEN, WPITEN and WPCREN bits. Always reads as 0

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x49434D (“ICM” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x49434D (“ICM” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x49434D (“ICM” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x49434D (“ICM” in ASCII).

Bit 0 – WPEN Write Protection Enable
See ICM Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x49434D (“ICM” in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x49434D (“ICM” in ASCII)

58.6.13 ICM Write Protection Status Register

Name: ICM_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status (Cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of ICM_WPSR.
1	A write protect violation has occurred since the last read of ICM_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

59. Classical Public Key Cryptography Controller (CPKCC)

59.1 Description

The Classical Public Key Cryptography Controller (CPKCC) is a Microchip macrocell that processes public key cryptography algorithm calculus in both GF(p) and GF(2^n) fields. The ROMed CPKCL, the Classical Public Key Cryptography Library, is the library built on the top of the CPKCC.

The Classical Public Key Cryptography Library includes complete implementation of the following public key cryptography algorithms:

- RSA (Rivest-Shamir-Adleman public key cryptosystem), DSA (Digital Signature Algorithm)
  - Modular Exponentiation with CRT up to 7168 bits
  - Modular Exponentiation without CRT up to 5376 bits
  - Prime generation
  - Utilities: GCD/modular Inverse, Divide, Modular reduction, Multiply, etc.
- Elliptic Curves
  - ECDSA GF(p) up to 521 bits
  - ECDSA GF(2^n) up to 571 bits
  - Point Multiply
  - Point Add/Doubling
  - Choice of the curves parameters so compatibility with NIST Curves or others
- Deterministic Random Number Generation (DRNG ANSI X9.31) for DSA

59.2 Product Dependencies

59.2.1 Power Management

The CPKCC is not continuously clocked. The CPCKCC interface is clocked through the Power Management Controller (PMC).

59.2.2 Interrupt Sources

The CPKCC has an interrupt line connected to the interrupt controller. Handling interrupts requires programming the interrupt controller before configuring the CPKCC.

59.3 Functional Description

The CPKCC macrocell is managed by the CPKCL available in the ROM memory of the device. The user interface of the CPKCC is not described in this section.

The usage description of the CPKCC and its associated library is provided in the application note “Using CPKCL Version 02.08.01.xx on SAMA7G5” (document no. DS00004438). This document is available under Non-Disclosure Agreement (NDA). Contact a Microchip Sales Representative for further details.

60. Security Module (SECUMOD)

60.1 Description

The Security Module (SECUMOD) provides secure functions to offer protection against voltage, temperature, frequency and mechanical attacks on the chip.

This module embeds the secure memories (5 Kbytes of SRAM and a 256-bit register bank) dedicated to the storage of sensitive data. These memories are scrambled with a programmable 32-bit key. The 5 Kbytes of SRAM use a non-imprinting mechanism.

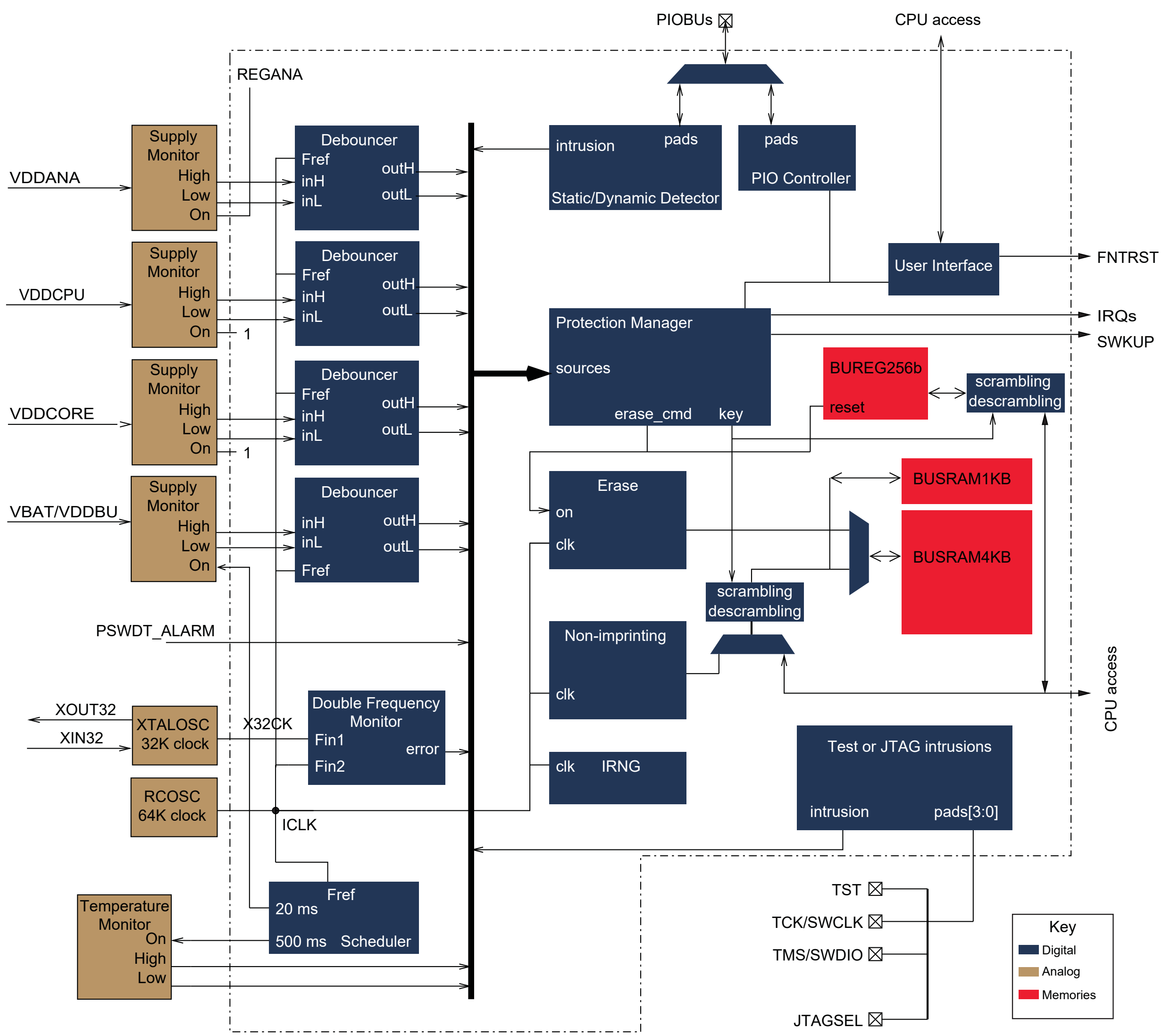
When a fault is detected, regardless of the source, a clear signal can be sent automatically to the secure memories and clear their contents.

60.2 Embedded Characteristics

- Voltage Monitoring
- Temperature Monitoring
- Frequency Monitoring
- PIO Controller Managing up to Four pads (PIOBU) with:
  - Two random dynamic signatures for mesh
  - Four external switch state change detectors
- JTAG Event Detector
- Test Entry Detector
- Programmable Secure Watchdog Alarm Detector
- Memory Erasing and Scrambling
- Operating Modes

60.3 Block Diagram

Figure 60-1. SECUMOD Block Diagram



60.4 I/O Lines Description

Table 60-1. I/O Lines Description

Name	Description	Type
PIOBU[0:3]	Parallel IO backup controller	I/O
TST	Test mode select. Must be connected to ground.	Input

...........continued		
Name	**Description**	**Type**
JTAGSEL	JTAG selection	Input
TCK/SWCLK	Test clock/serial wire clock	Input
TMS/SWDIO	Test mode select/serial wire	I/O
XIN32	Slow clock oscillator input	Input
XOUT32	Slow clock oscillator output	Output

60.5 Product Dependencies

60.5.1 Interrupt Sources

The Security Module provides two interrupt lines, each connected to one of the internal sources of the interrupt controller. Using these interrupts requires the interrupt controller to be programmed first. Interrupt lines must not be used in Edge-sensitive mode.

The first interrupt line (SECURAM, peripheral identifier 18) is used to signal violations of backup memory access rights, or to signal end of erase (automatic or software erase).

The second interrupt line (SECUMOD, peripheral identifier 17) is shared by all the protection mechanisms except backup memories access right violations signaling, or end of erase (automatic or software erase) signaling.

60.6 Functional Description

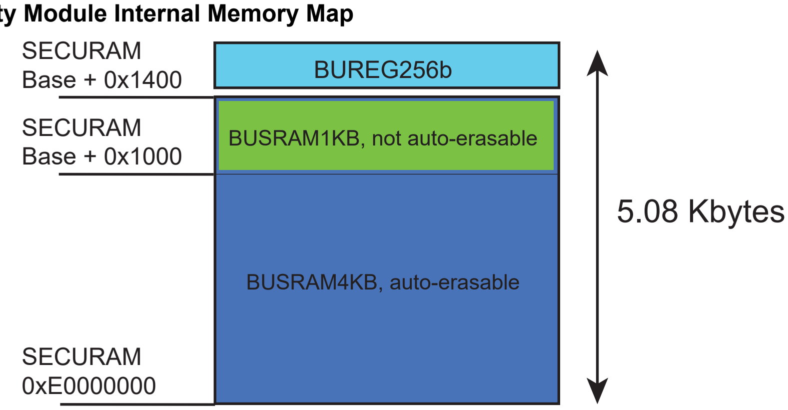
60.6.1 Memory Mapping

The Security Module embeds 5 Kbytes of SECURAM split in two parts: the lower 4 Kbytes are erased in case of intrusion (BUSRAM4KB) while the upper 1 Kbyte is never erased (BUSRAM1KB). A 256-bit register bank is available as an additional memory and is totally erased in case of intrusion (BUREG256b).

All memories support 8-bit, 16-bit and 32-bit access sizes.

To optimize power consumption, the transfers between the processor and these memories are slower than for other memories in the device.

Figure 60-2. Security Module Internal Memory Map



60.6.2 Scrambling Keys

The secure memories (BUSRAM4KB, BUSRAM1KB and BUREG256b) are scrambled. The scrambling key can be modified through the Scrambling Key register (SECUMOD_SCRKEY). The scrambling key reset value differs between devices.

Scrambling can be disabled in the Control register (SECUMOD_CR).

60.6.3 Protection Mechanisms

60.6.3.1 Protection Manager

The Protection Manager is used to centralize all alarms coming from the different monitors.

The Protection Manager implements an automaton that processes a memory erase sequence if the memory is not empty.

In Normal mode each available alarm described in this document can be configured as follows:

- Disabled
- Trigger an automatic memory erase sequence
- Trigger an IRQ interrupt
- Trigger an automatic memory erase sequence and an IRQ interrupt

In Backup mode each available alarm described in this document can be configured as follows:

- Disabled
- Trigger an automatic memory erase sequence
- Trigger a SWKUP wake-up signal
- Trigger an automatic memory erase sequence and a SWKUP wake-up signal

The software knows an alarm has been triggered by reading the Status register (SECUMOD_SR), by enabling the IRQ interrupt or by enabling the SWKUP wake-up. The software then detects the source of the alarm and acts accordingly. The software erases the memory if not done automatically. The software triggers a memory erase sequence using SECUMOD_CR.SWPROT.

The Protection Manager can also send:

- an IRQ interrupt signal (only in Normal mode)
- an SWKUP wake-up signal (only in Backup mode).

As soon as an alarm is detected, the corresponding bit is set in the Status register (SECUMOD_SR). The only way to clear this bit is to set it in the Status Clear register (SECUMOD_SCR).

If a clear of the secure memories content has been performed by the automaton, an ERASE_DONE flag is set to indicate that the secure memories content is not valid anymore. While the secure memories are erased, write accesses have no effect and read accesses return a static and invalid value (except for BUSRAM1KB).

60.6.3.2 Test and JTAG Pin Monitor

This monitor controls the activity on the TST, TMS/SWDIO, TCK/SWCLK and JTAGSEL pads, and on some processor pins.

If the TST pin is seen high on two consecutive ICLK clock cycles, an alarm is sent to the Protection Manager in order to start the Erase sequence.

If the JTAGSEL pin state changes either to Debug or Boundary Scan mode, an alarm is generated. Refer to EmbeddedICE^{TM} in the section "Debug and Test" for the polarity of the JTAGSEL pin.

If the TMS/SWDIO pin is seen high twice and seen low twice in less than 10 consecutive TCK/SWCLK clock cycles, the TCK alarm is sent to the Protection Manager in order to start the Erase sequence. This detection is used to detect some JTAG commands being issued from an external debugger.

If the DBGACK pin of the processor is seen high (signaling that the core entered Debug mode), an alarm is sent to the Protection Manager, provided SECUMOD_JTAGCR.PROC_DEBUG_MON=1.

60.6.3.3 PIO Backup Controller

The Security Module includes a PIO Controller powered by VBAT which handles the 4 PIOBU I/O pins.

Each I/O line is controlled by the PIO Controller and each pin can be configured to be driven. This is done by writing in the corresponding SECUMOD PIO Backup register (SECUMOD_PIOBUx). When SECUMOD_PIOBUx.OUTPUT is at ‘0’, the corresponding I/O line is used as an input only. When this bit is at ‘1’, the corresponding I/O line is driven by the PIO Backup Controller.

60.6.3.3.1 Output Mode

When SECUMOD_PIOBUx.OUTPUT is set, the level driven on an I/O line can be determined by setting or clearing SECUMOD_PIOBUx.PIO_SOD (Set Output Data). The value of this bit represents the data driven on the corresponding I/O line.

60.6.3.3.2 Input Mode

The level on an I/O line can be read from SECUMOD_PIOBUx.PIO_PDS (Pin Data Status). This bit indicates the level of the I/O line regardless of its configuration, whether as an input or driven by the PIO Controller.

60.6.3.3.3 Static Intrusion Detectors and Programmable Internal Pull-up/Pull-down

Intrusion detectors can be placed around the system to deter any intrusion attempt. This requires the corresponding I/O lines to be configured as inputs (SECUMOD_PIOBUx.OUTPUT='0'). Each pair of I/Os can be configured in Static or Dynamic Intrusion Detection mode. By default, intrusion detection is in Static Intrusion Detection mode.

60.6.3.3.4 Static Intrusion Detection

The detectors can be configured to detect either the rising edge or the falling edge on switches via SECUMOD_PIOBUx.SWITCH.

As an example, a detector can consist of a normally-closed switch which sends a zero signal to the Protection Manager. When an intrusion attempt occurs, the switch state changes to an open position. The debounce filter waits until an intrusion has been detected for a programmable continuous period to send an alarm signal to the Protection Manager. This prevents erroneous intrusion detections.

60.6.3.3.5 Internal Pull-Up/Pull-Down

An internal pull-up or pull-down (around 100 kΩ) can be connected by configuring SECUMOD_PIOBUx.PULLUP.

Configuring this field with a pull-up or pull-down value activates the corresponding pull-up/pull-down permanently.

Note: Internal pull-ups are connected at Reset state.

60.6.3.3.6 Scheduled Pull-Up/Pull-Down

In order to reduce the power consumption on the VBAT power supply, all activated pull-ups/pull-downs can be scheduled by following the steps below:

1. Activate the required pull-up/pull-down.
2. Measure the level on the PIOBUx pin.
3. Deactivate the pull-up/pull-down.

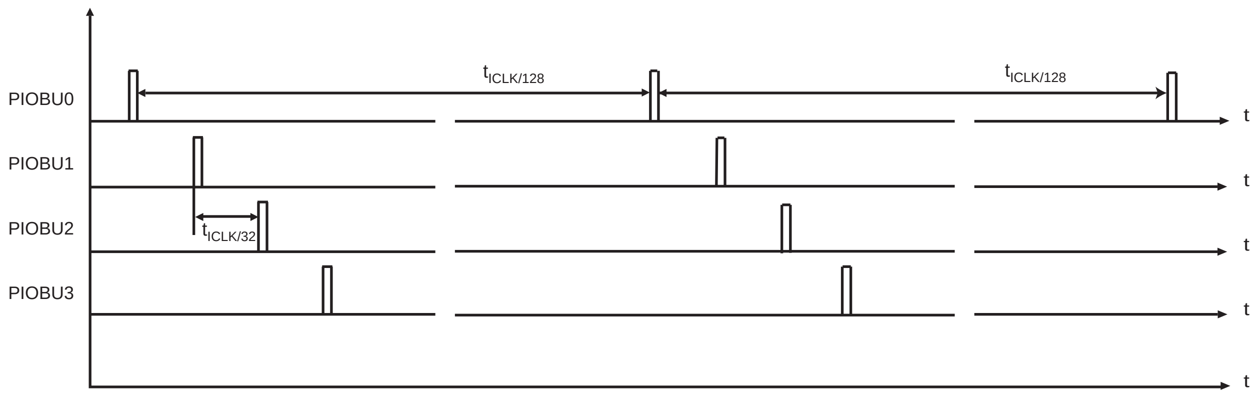
Scheduling is enabled by setting SECUMOD_PIOBUx.SCHEDULE.

Note: This feature is only effective if SECUMOD_PIOBUx.PULLUP= '1' or '2', indicating that a pull-up or a pull-down is connected.

60.6.3.3.7 Debouncing Time

The debouncing time is common to all I/Os. The principle is presented in the figure below. A period ($f_{ICLK}/2$) is allocated to each I/O. During that period, if SECUMOD_PIOBUx.SCHEDULE is set and if a pull-up/pull-down is needed (PULLUP≠0), the pull-up/pull-down is activated, the level is measured and the pull-up/pull-down is deactivated. Otherwise, only the level is measured. Measurement is performed at the end of the allocated period. Inter pulse delay is set via SECUMOD_PIOBUx.PIOBU_RFV.

Figure 60-3. Schedule Principle



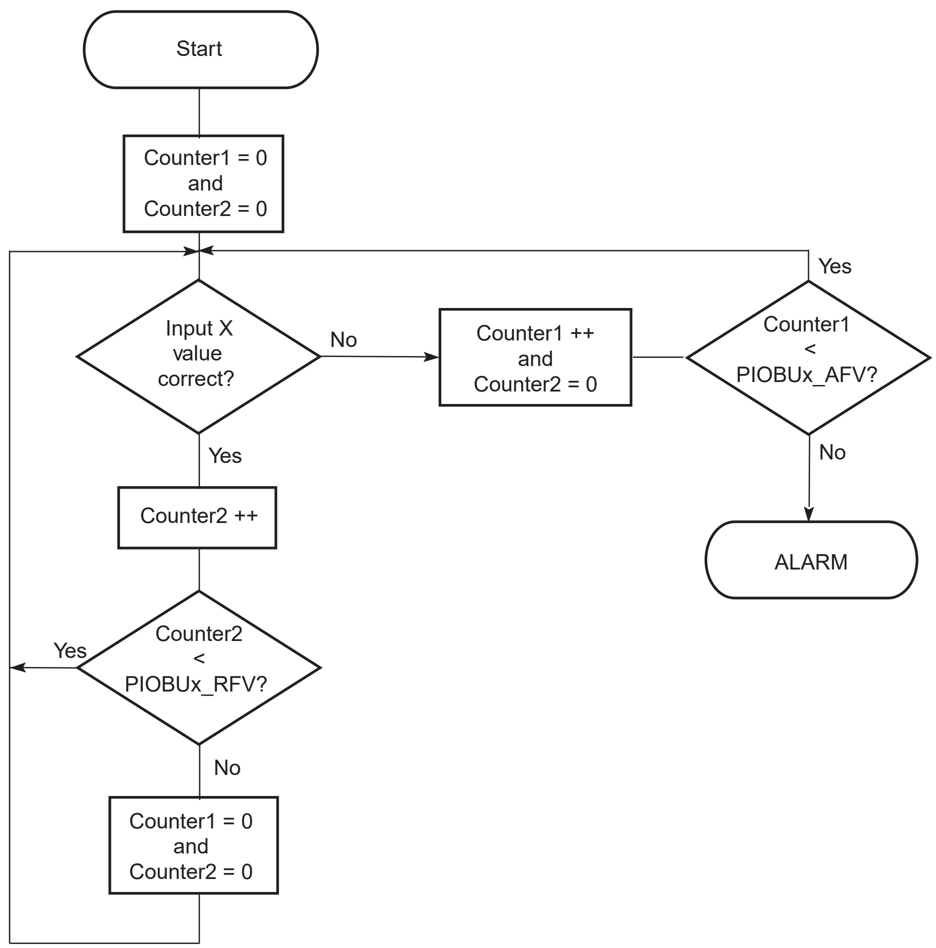
60.6.3.3.8 PIOBUx Alarm Filtering in Static Mode

Filtering the PIOBUx alarm detection is done by programming SECUMOD_PIOBUx.PIOBU_AFV with the value that corresponds to the maximum counter value. See SECUMOD_PIOBUx. The steps are as follows:

1. A 9-bit counter is incremented each time the value present on the corresponding input is not the expected one.
2. An alarm is sent to the Protection Manager if the counter value reaches the value programmed in PIOBU_AFV.

The 9-bit counter is reset only if the value present on the input is correct and stable for a continuous programmable period defined by SECUMOD_PIOBUx.PIOBU_RFV. Another 9-bit counter is necessary for that operation. See the figure PIOBUx Alarm Filtering Principle.

Figure 60-4. PIOBUx Alarm Filtering Principle



At reset, the debouncers are not activated (PIOBU_AFV and PIOBU_RFV=0), which implies that no alarm can be generated.

Once both PIOBU_AFV and the PIOBU_RFV are programmed, the corresponding protection is activated and a clear signal is generated automatically when an intrusion is detected.

Instead of clearing the secure memories content, an interrupt or a wakeup signal can be generated. To do so, disable the protection in the Normal Mode Protection register (SECUMOD_NMPR) and configure the Normal Interrupt Enable Protection register (SECUMOD_NIEPR).

Note: If the Normal Mode Protection/Backup Mode Protection registers are not hidden, their configuration has priority over the debouncer activation in the PIOBUx configuration registers. This means that clear signal generation is enabled/disabled in those two registers. Setting the PIOBU_AFV and PIOBU_RFV fields configure the debouncer sensitivity and does not generate any clear signal when an intrusion is detected.

60.6.3.3.9 Dynamic Intrusions

Principle

The internal random number generator (IRNG) is used to generate dynamic signatures through the I/O pins split in pairs. For each pair, a signature is transmitted on the even-indexed I/O, and must be externally fed back on the odd-indexed I/O. The following order must be followed for an external connection:

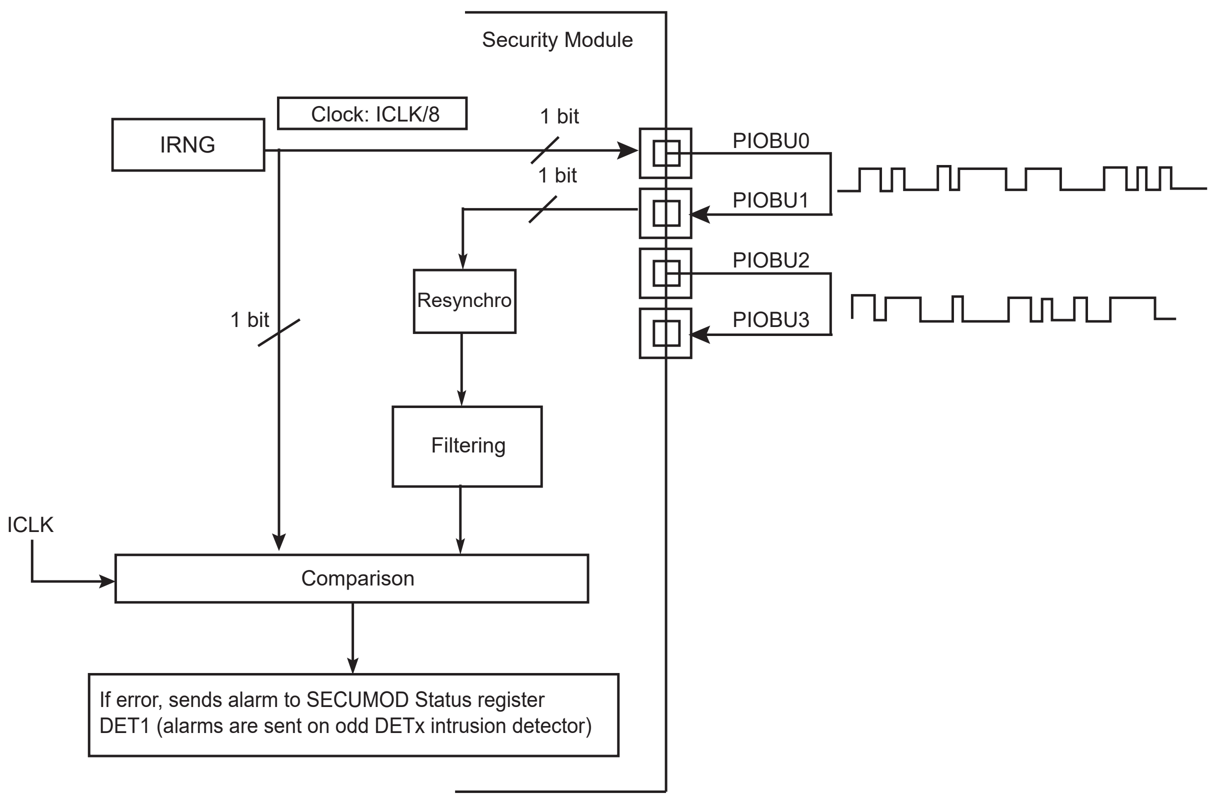
1. PIOBU0 → PIOBU1
2. PIOBU2 → PIOBU3

A dynamic signature is made of random patterns (one pattern = one high state followed by one low state). The duration of stable states is a multiple of an internal signature clock.

The signatures are unique and incompatible. This connection order is mandatory.

The figure below provides an overview of the dynamic detection system.

Figure 60-5. Dynamic Detection System



Enabling Dynamic Detection

Dynamic detection is enabled by writing a ‘1’ to SECUMOD_PIOBUx.DYNSTAT (active high).

Dynamic Detection Intrusion and Static Detection Intrusion modes are incompatible. Static mode must be disabled by writing a '0' to PIOBU_AFV and PIOBU_RFV of SECUMOD_PIOBUx for a PIO configured in Dynamic Detection Intrusion mode.

Tuning Dynamic Detection
The signature refresh rate (signature clock) and the sensitivity of the detection can be tuned in the Dynamic Signatures Tuning register (SECUMOD_DYSTUNE). The tuning is common to all signatures, but each pair has its own independent detection mechanism.

For each pair, once enabled, the detection mechanism increments an error counter after each mismatching pattern detected on the receiver pin. Concurrently, consecutive matching patterns increment another counter. When this counter reaches the threshold programmed in SECUMOD_DYSTUNE.RX_OK_CORREL_NUMBER, the error counter is reset. When the error counter reaches the threshold programmed in SECUMOD_DYSTUNE.RX_ERROR_THRESHOLD, an alarm is generated. The overlapped behavior of these two counters results in the ability of the system to tolerate a pattern error ratio below which no alarm is generated. This ratio derives from the programmed fields though the following formula:

Tolerated Error Ratio = (RX_ERROR_THRESHOLD -1) / RX_OK_CORREL_NUMBER

A great flexibility is given in the programming interface to let the user find the best setting for the application.

The detection algorithm state machine is illustrated in the figure Dynamic Detection Mechanism State Machine. The terms used are as follows:

- Error counter = NBERR
- Consecutive matching patterns counter = PATTERN_OK
- RX_OK_CORREL_NUMBER field = OK_THR
- RX_ERROR_THRESHOLD field = ERR_THR

Various scenarios of state machine operation are illustrated in the figures Dynamic Signature Ongoing, No Error, Dynamic Signature Ongoing, One Error, Dynamic Signatures: Resetting Error Counter After Receiving a Sufficient Number of Good Patterns and Dynamic Signatures: Raising Alarm After Receiving a Sufficient Number of Bad Patterns.

Figure 60-6. Dynamic Detection Mechanism State Machine

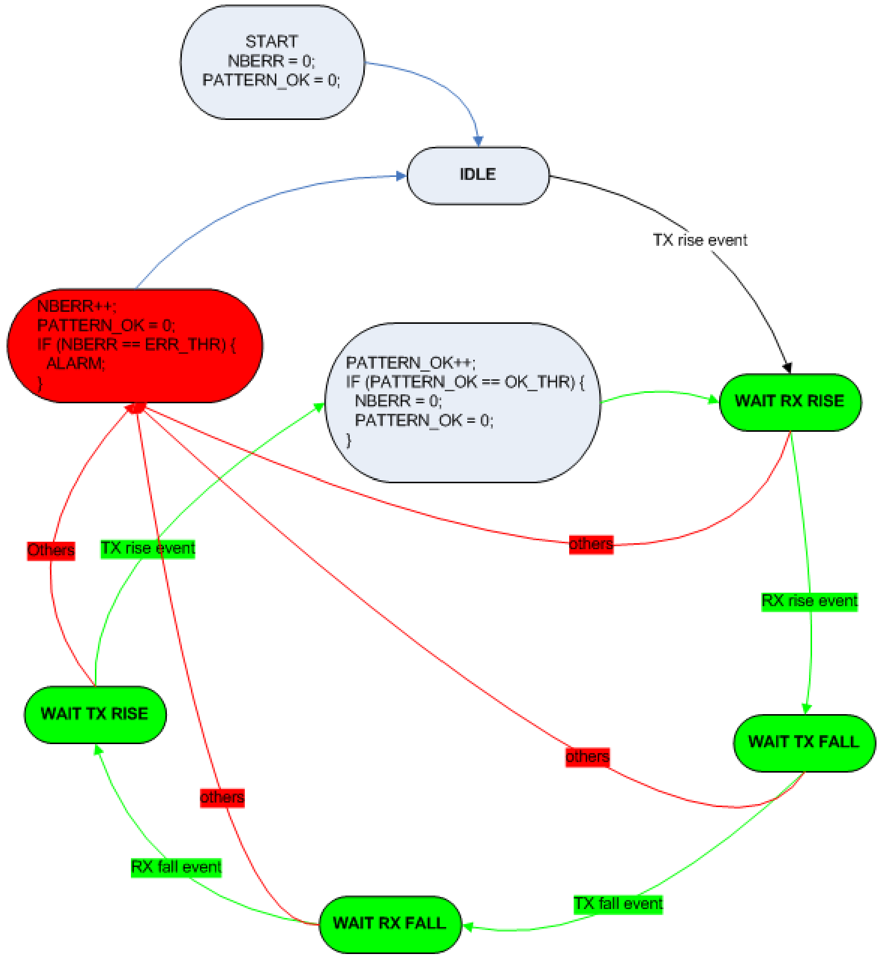


Figure 60-7. Dynamic Signature Ongoing, No Error

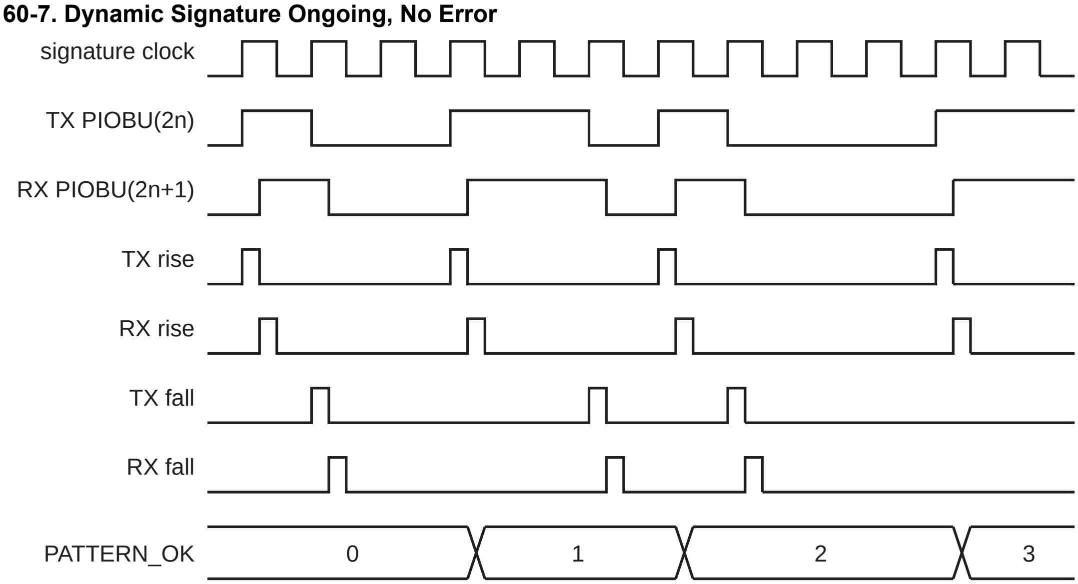


Figure 60-8. Dynamic Signature Ongoing, One Error

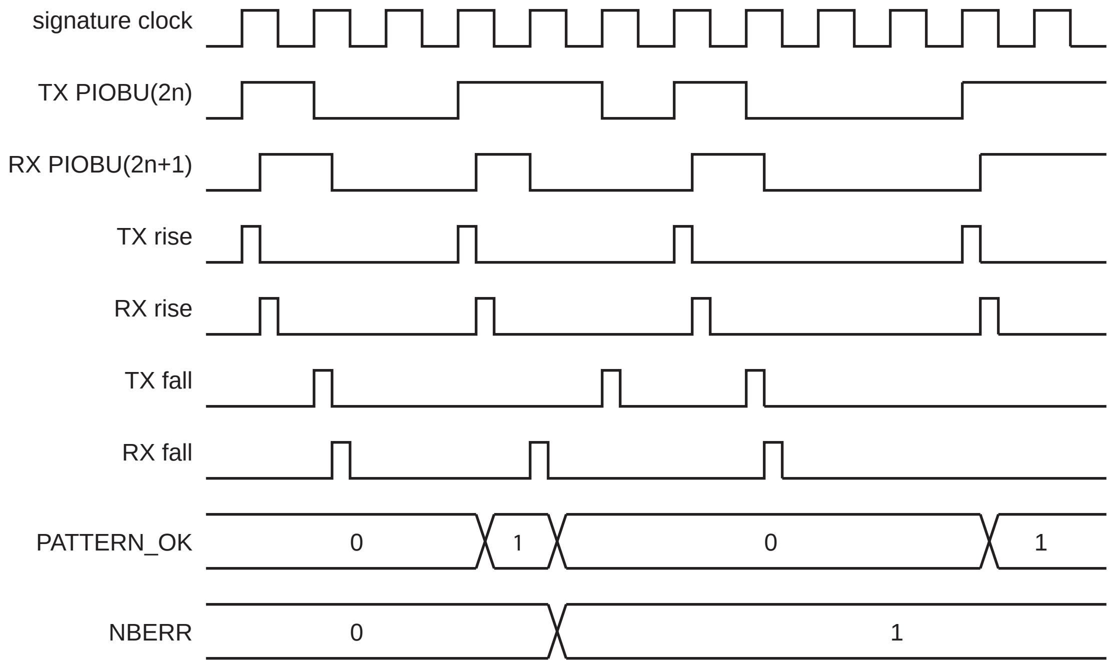


Figure 60-9. Dynamic Signatures: Resetting Error Counter After Receiving a Sufficient Number of Good Patterns

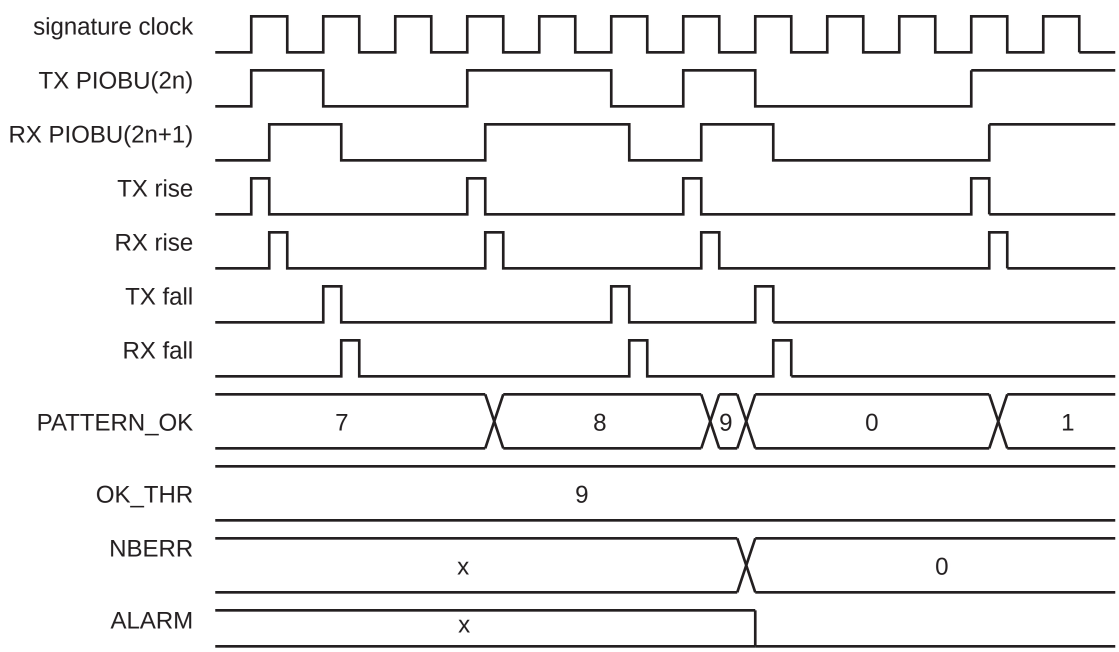
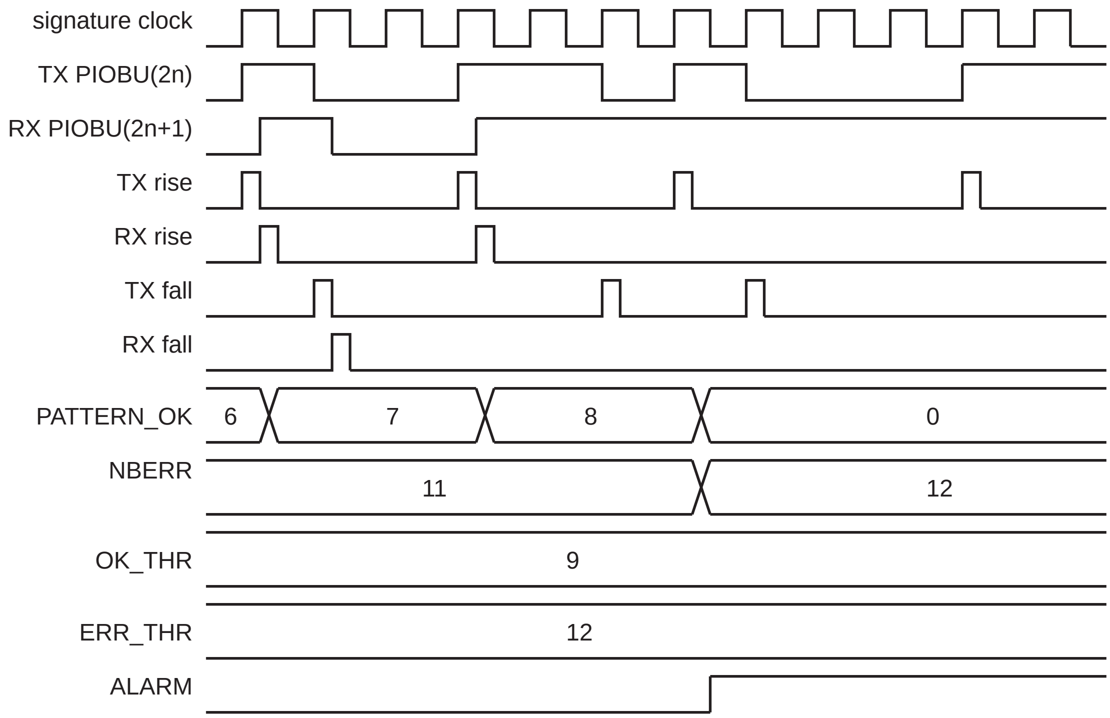


Figure 60-10. Dynamic Signatures: Raising Alarm After Receiving a Sufficient Number of Bad Patterns



External Filtering in Dynamic Mode

A majority vote filter is used to adapt filter cut-off sensitivity (preventing a false detection due to any spurious pulses).

The filter can be configured through SECUMOD_PIOBUx.FILTER3_5. All pulses less than $t_{ICLK}/8$ or $t_{ICLK}/4$ are filtered when FILTER3_5 is low (default) or high, respectively.

60.6.3.4 JTAG Prevention

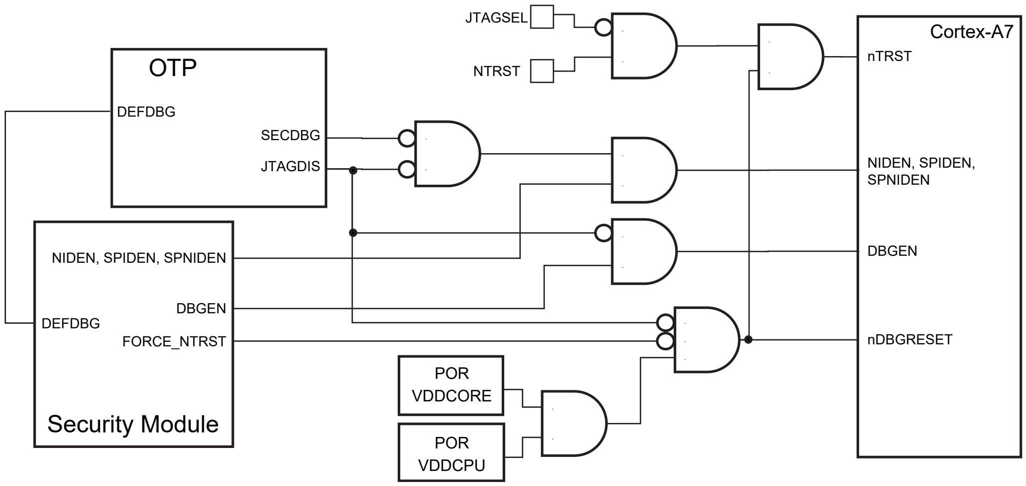
60.6.3.4.1 Debug Interface Access Prevention

The Security Module can be used to block access to the system through the Arm processor Debug Access Port interface. This feature is implemented via SECUMOD_JTAGCR, which enables assertion of the nDBGRESET reset input of the debug interface. Writing SECUMOD_JTAGCR.FNTRST (Force NTRST) to ‘1’ in this register prevents any activity on the TAP (Test Access Port) controller.

Several protections have been implemented in order to prevent JTAG Debug modes.

FNTRST also locks the boundary scan when set.

Figure 60-11. JTAG Protection Principle



60.6.3.4.2 Physical Prevention for JTAG Debug

A physical protection has been implemented for JTAG debug. The TMS/SWDIO, TCK/SWCLK and TDI signals are connected directly to the boundary TAP controller, but they can be inhibited on the path to the Debug Access Port (DAP). Thus, to disable the DAP for debug purposes, the JTAGDIS field must be configured in the customer OTP area.

Programming this field prevents JTAG debug irreversibly.

60.6.3.4.3 Physical Restrictions for JTAG Debug Mode

Invasive and non-invasive debug modes are controlled by four input pins of the Debug Access Port: DBGEN, SPIDEN, NIDEN and SPNIDEN.

To restrict the debug to non-secure software parts only, the SECDBG field must be configured in the customer OTP area.

Programming this fields prevents JTAG secure debug irreversibly, but does not lock non-secure debug.

60.6.3.4.4 Software Prevention for JTAG Debug

It is possible to prevent JTAG Debug accesses by forcing the reset signal of Debug Access Port by software.

While the reset signal is maintained low, the JTAG Debug interface cannot be used. To maintain the Debug Access Port in Reset state, set SECUMOD_JTAGCR.FNTRST. In this case, Boundary JTAG is also disabled.

60.6.3.4.5 Software Restrictions for JTAG Debug Mode

Setting SECUMOD_JTAGCR.PROC_DEBUG_MODE sets the processor inputs DBGEN, SPIDEN, NIDEN and SPNIDEN to the appropriate level in order to allow different debug permission levels. See SECUMOD_JTAGCR for more information.

The reset value of SECUMOD_JTAGCR can be chosen as ‘full debug’ or ‘no debug’, depending on the programming of the SECDBG field in the customer OTP area.

Programming this field sets the default debug level after reset irreversibly.

Note that the OTP settings described above have priority over software settings, so SECUMOD_JTAGCR is not able to restore modes which are disabled through OTP. As an example, if the SECDBG field is programmed, allowing secure debug in SECUMOD_JTAGCR has no effect.

60.6.3.5 Voltage Monitors

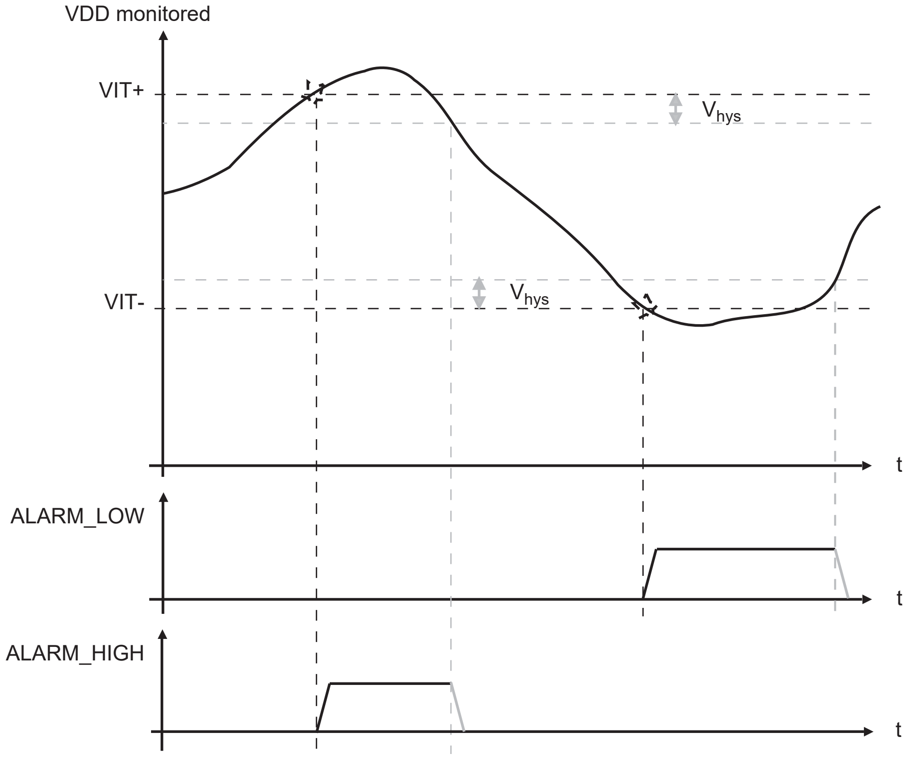
Voltage monitors verify the input voltage and send an error signal when the measured voltage is out of range.

Two types of alarm are generated:

1. High threshold overtaking
2. Low threshold undertaking

In Normal mode, i.e., when the chip is fully powered, voltage measurements are performed on VDDCORE, VDDCPU and VBAT. VDDCORE, VDDANA and VDDCPU are continually monitored, whereas VBAT and VDDBU measurements are scheduled in order to reduce power consumption on the VBAT power supply. When Backup mode is active, only VBAT is monitored.

Figure 60-12. Voltage Monitor Principle



60.6.3.5.1 VDDCORE Voltage Monitor Characteristics

See the figure Voltage Monitor Principle and refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

60.6.3.5.2 VDDCORE Alarm Filtering

The debouncer sensitivity is fully programmable using the required VDDCORE_DBTV value in the VDDCORE Filter register (SECUMOD_VCOREFR).

The debouncing time is calculated using the following formula:

Debouncing Time = VDDCORE_DBTV/f_{TD_SCLK}

Refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

60.6.3.5.3 VDDCPU Voltage Monitor Characteristics

The VDDCPU Monitor voltage range can be adapted to two operating modes (600 MHz Max or 800 MHz Max) through SECUMOD_GPSBR.SMCPURANGE.

See the figure Voltage Monitor Principle and refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

60.6.3.5.4 VDDCPU Alarm Filtering

The debouncer sensitivity is fully programmable using the required VDDCPU_DBTV value in the VDDCPU Filter register (SECUMOD_VCPUFR).

The debouncing time is calculated using the following formula:

DebouncingTime = VDDCPU_DBTV/f_{TD_SLCK}

Refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

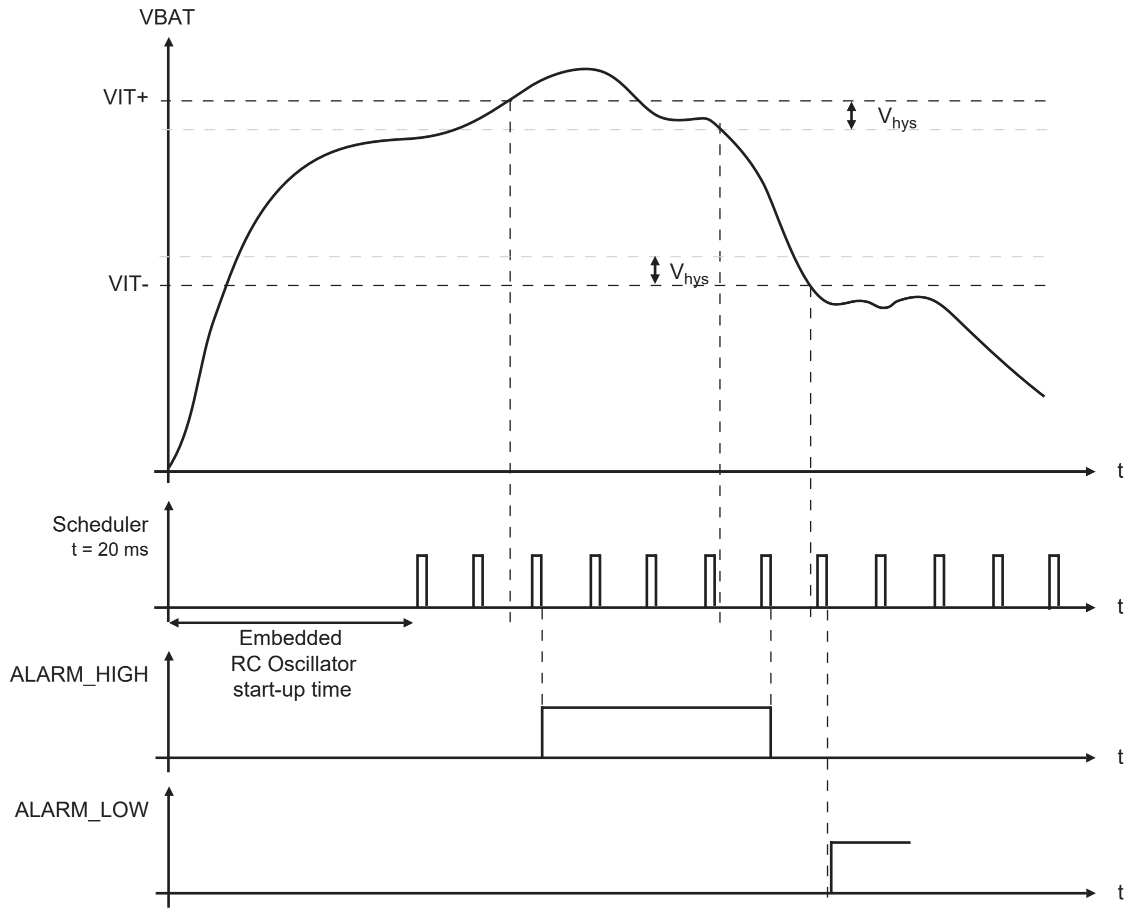
60.6.3.5.5 VBAT and VDDBU Alarm Filtering

The VBAT and VDDBU alarm detection can be filtered by programming the VBAT Filter register (SECUMOD_VBUFR). The sequence is as follows:

1. Each 20 ms, a VBAT (or VDDBU) measure is performed.
2. If the voltage monitor detects an error, an alarm is sent to the protection unit if the counter value reaches the value programmed in the field Filter Value (SECUMOD_VBUFR.VBATFV) (maximum programmable value = 7).
3. The counter is cleared if the voltage monitor detects no further error.

Refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

Figure 60-13. Voltage Monitor Hysteresis



60.6.3.5.6 VDDANA Alarm

The VDDANA alarm detection is always on. REGANA high and low voltage alarms are respectively REGANA_HI and REGANA_LO bits in SECUMOD_ASR.

Refer to Voltage Monitoring in the section "Electrical Characteristics" for more information.

60.6.3.6 Temperature Monitor

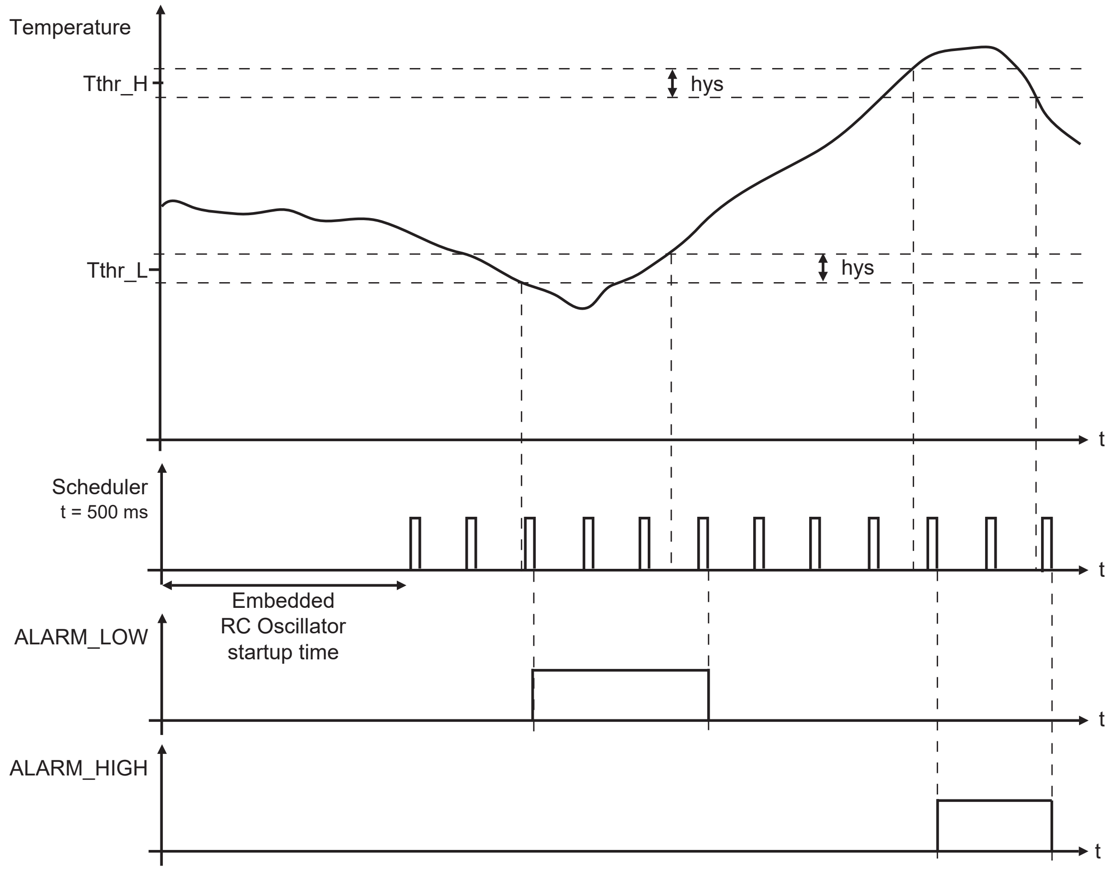
Each 500 ms, the temperature measure is performed and the temperature monitor warns the protection unit when the temperature is out of range.

Two types of alarm are generated:

- High threshold overtaking
- Low threshold overtaking

The temperature monitor high threshold can be set to two different temperatures by programming SECUMOD_GPSBR.TSRANGE. OTP bits can be used to allow or prevent this reconfiguration and select a permanent setting when reconfiguration is not allowed.

Figure 60-14. Temperature Monitor Hysteresis



Refer to Temperature Monitoring in the section "Electrical Characteristics" for more information.

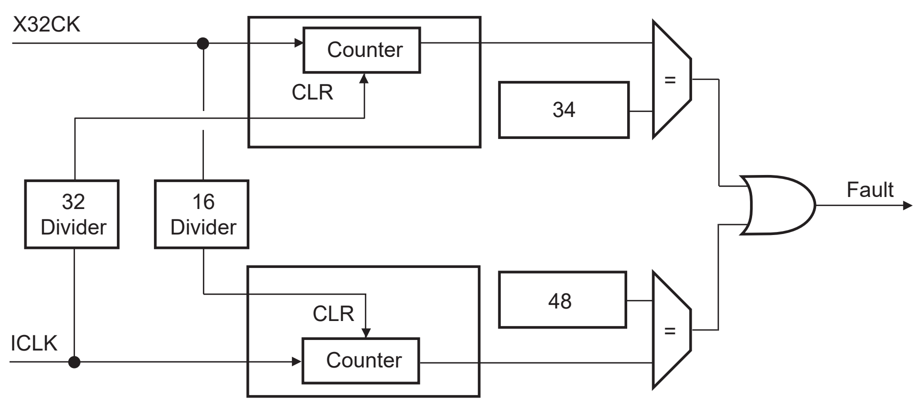
60.6.3.7 Double Frequency Monitor

The double frequency monitor checks the two points below, assuming that the frequency of the 32.768 Crystal Oscillator (X32CK) is lower than the frequency of the 64 kHz RC Oscillator (ICLK) (typically f_{ICLK} = 64 kHz and f_{X32CK} = 32.768 kHz):

1. The number of ICLK periods in 16 X32CK periods must not exceed 48.
2. The number of X32CK periods in 32 ICLK periods must not exceed 34.

These values are calculated based on the extreme values of the frequency ranges.

Figure 60-15. Double Frequency Monitor Principle



60.6.4 Non-Imprinting

60.6.4.1 Principle

To avoid data remanence in the memory, the 5 Kbytes of SRAM are periodically reverted. The following two mechanisms are used:

1. When enabled, the non-imprinting circuitry periodically reads-inverts-writes each word of the SRAM. The inversion state of each word is stored in the memory.
2. When the processor writes data into memory, it is pseudo-randomly chosen so that either the right or the reverted data is written. The selected inversion state is stored at the same time.

The full non-imprinting process needs 2560 ICLK cycles to complete.

60.6.4.2 Conditions Enabling or Disabling the Non-Imprinting Process

The non-imprinting mechanism is disabled by default and must be authorized by setting the bit NIMP_EN in the Control register (SECUMOD_CR).

To have the least possible impact on application execution, the non-imprinting can start only if at least one of the following conditions is true:

- The core supply domain is off (i.e., the Security Module is in Backup mode).
- The processor is in Idle state (“Wait For Interrupt” or “Wait For Event”).
- The peripherals RESET is asserted from the System Controller user interface.

When one of these conditions is true, no system access can occur to the memory and the non-imprinting access priority is granted.

The RTC is used to trigger the periodic execution of non-imprinting, once a day between 3 a.m. and 4 a.m.

If none of the three conditions above are met, the process does not start. If one or more conditions are met, the process starts and stops when all memory is reverted. Then, the process is inhibited until the day after.

If the process is running and all of the three conditions above become false, the non-imprinting process is stopped as fast as possible, in a maximum time of 3 ICLK clock cycles.

If an intrusion occurs and triggers an erase process, the non-imprinting process stops cleanly in one period of ICLK clock and gets the access priority on the memory until the erase process completes.

As long as none of the three conditions returns true, the non-imprinting process is inhibited.

60.6.4.3 Granting System Access After Going Out of Imprinting Conditions

As described in Conditions Enabling or Disabling the Non-Imprinting Process, when the non-imprinting enable condition becomes false, the process is stopped in a few ICLK clock cycles (if it was running).

During this time, the processor cannot access the memory and must wait for the automatic end of process. Due to the frequency difference between the processor clock and the ICLK clock, adding thousands of Wait states on the system bus (which would lock the application for a long time) would be inappropriate. It was decided instead to add a register in the Security Module user interface, SECUMOD_RAMRDY, which returns the availability status of the memory.

The following formula gives the maximum number of processor cycles to wait before RAMRDY is set.

CPU cycles $\leq 3 \times f_{CPU}/f_{ICLK}$

60.6.5 Erasing Secure Memories

60.6.5.1 BUSRAM4KB Erase Sequence

The BUSRAM4KB Erase sequence is activated by the clear signal. The Erase sequence processes 64-bit words and is necessary to degrade the BUSRAM4KB content as fast as possible.

BUSRAM4KB is erased only if its contents have been modified since the last erase, or if it is the first erase request since the backup reset was released. This mechanism prevents recurring erases that consume power unnecessarily, in particular when the backup area is powered only by the backup battery. See SECUMOD_SYSR for software management of the erase. Even if the physical erase of the memory is not done due to an empty memory, the intrusion flags are stored in the Status register.

The Erase automaton performs 64-bit accesses, allowing “as fast as possible” erase.

During the Erase sequence, the upper 1 Kbyte of memory (BUSRAM1KB) is still accessible by the system.

To enhance security, the Erase sequence fills the memory with random values.

The erase sequence takes 64 cycles of ICLK for a partial erase and 512 cycles of ICLK for a full erase.

60.6.5.2 BUREG256b Erase Sequence

In parallel to the BUSRAM4KB Erase, the BUREG256b register bank is erased immediately.

BUREG256b is erased only if its contents have been modified since the last erase, or if it is the first erase request since the backup reset was released. Even if the physical erase of the memory is not done due to an empty memory, the intrusion flags are stored in the status register.

BUREG256b is always reset after a VBAT power-up.

These registers are seen as zero after reset or after an erase.

60.6.5.3 During and After BUSRAM4KB and BUREG256b Erase Sequence

Some flags can be read to know the real-time erase state of the memories. On completion of the erase sequence, the SECURAM ID interrupt line is asserted.

For more information on software management, see SECUMOD_SYSR.

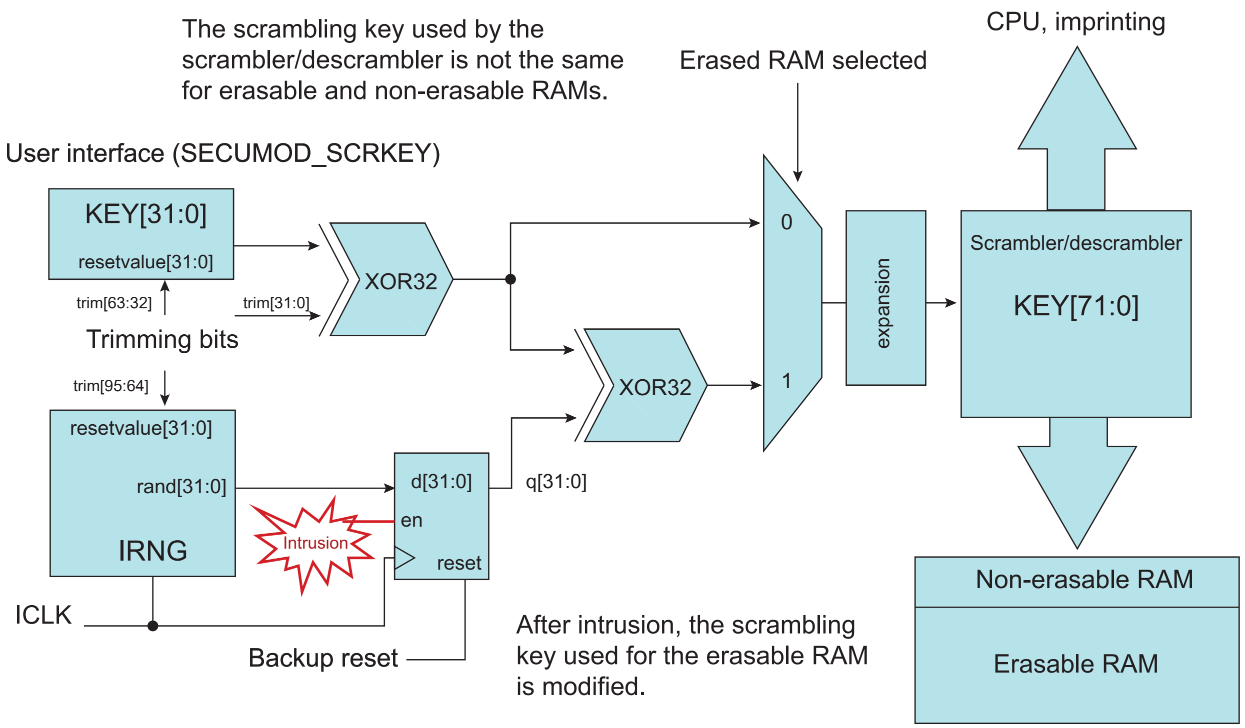
After an Erase sequence, the memories are considered empty. The first write access to the memories indicates that they are not empty anymore and thus that other physical erases can be performed.

After reset or after an erase, it is recommended to reinitialize the full BUREG256b register bank to avoid unexpected values appearing on uninitialized addresses when starting to write into this memory. This can be achieved by writing zeros to unused locations.

60.6.5.4 Scrambling Key Protections

To enhance the security level, the scrambling key applied to the erasable memories is modified immediately before and also just after the physical erase sequence. Note that in case of empty memory, despite the fact that the sequential erasing process is not launched as previously described, the scrambling key of erasable memories is modified. The global key, which is configured in SECUMOD_SCRKEY, is not modified since the non-erasable part of the memory must be preserved. The figure below illustrates the chosen implementation.

Figure 60-16. Scrambling Key



The use of trimming bit XORs differentiates parts in the key final values reaching the scrambler/descrambler.

The key used for the BUREG256b scrambler/descrambler is derived from the BUSRAM4KB key and thus benefits from the same protection.

60.6.6 Operating Modes

The Security Module is supplied by the VBAT power supply. It is not possible to program the Security Module if VBAT is not present.

The Security Module macrocell is able to operate in two different modes:

- When all supplies are present and can be monitored, the Security Module can be switched to Normal mode.
- Otherwise, the Security Module must be in Backup mode.

Note: After a power-up reset, the Security Module is in Backup mode.

The mode is selected in SECUMOD_CR by setting either the NORMAL bit or the BACKUP bit.

Note: The user must set the BACKUP bit to enter Backup mode prior to shutting off the VDDCORE power supplies.

In both modes, the user can enable or disable a protection by writing in the corresponding Mode Protection register. See Activation or Deactivation of Protections for more information.

60.6.6.1 Normal Mode

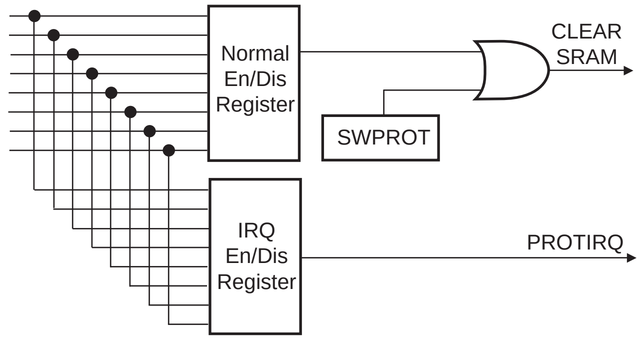
When running in Normal mode, the Security Module monitors all the sensors. Some of the sensors are out of the backup power domain, for example JTAG and core voltage.

When detecting an error, the Protection Manager sends a CClear signal to the automaton, which starts the secure memories Erase sequence if the memory is not empty.

The user can generate an interrupt signal (IRQ) without starting the Erase sequence. That choice is defined by writing either in SECUMOD_NMPR or in SECUMOD_NIEPR.

The user can generate a software clear of BUSRAM4KB and BUREG256b by setting SECUMOD_CR.SWPROT. Setting this bit has no effect during the automatic Erase sequence. After completion of the erase process, the SECURAM ID interrupt line triggers an interrupt to the processor.

Figure 60-17. Normal Mode



60.6.6.2 Backup Mode

When running in Backup mode, only the backup power domain is monitored by the Security Module.

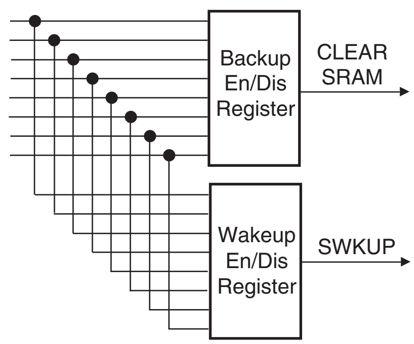
When detecting an error, the Protection Manager sends a clear signal to the automaton, which starts the secure memories Erase sequence if the memory is not empty.

The user can generate a wake-up signal (SWKUP) to the Shutdown Controller instead of clearing the secure memories content immediately. This choice is defined by writing in SECUMOD_BMPR or in SECUMOD_WKPR.

Refer to the Shutdown Mode register (SHDWC_MR) in the section “Shutdown Controller (SHDWC)” for details on wake-up event management.

Note that the Security Module Backup mode differs from the Shutdown Controller Backup mode in that the Shutdown Controller Backup mode is only related to the VDDCORE supply presence, while the Security Module Backup mode is not.

Figure 60-18. Backup Mode



60.6.6.2.1 Backup Mode Configuration

To switch to Backup mode:

1. Configure SECUMOD_BMPR or SECUMOD_WKPR to enable the protections that will be required when Backup mode is activated.
2. Set SECUMOD_CR.BACKUP to switch to Backup mode.
3. Shut down all power supplies except VBAT.

The Backup mode protection is now enabled and the VDDCORE voltage monitors are disabled. This avoids detection of an error signal coming from one of these monitors and a subsequent automatic clear of BUSRAM4KB and BUREG256b.

60.6.6.3 Automatic Backup Mode

The Security Module can switch automatically to Backup mode through a specific protocol involving EXT_IRQ and NRST product pins. This protocol ensures that the Security Module does not receive core alarms when VDDCORE is removed. This may occur if VDDCORE is dropping too fast with respect to the time needed by the software to switch to Backup mode.

The only way to avoid this is to anticipate this event soon enough with an external event signaling that the VDDCORE is going to be removed. The EXT_IRQ has been chosen for the external event.

In this protocol, the VDDCORE must be removed only while the NRST pin is low and user reset is enabled in the system controller.

The Automatic Backup mode must first be enabled by writing a '1' to SECUMOD_CR.AUTOBKUP.

If a low-level transition on the EXT_IRQ pin is detected while the Automatic Backup mode is on, the Security Module memorizes the current mode setting and switches to Backup mode. It takes up to 5 TD_SLCK cycles to complete.

While the Automatic Backup mode is set, all intrusions generated out of the backup domain are ignored by the Security Module.

If a high-level transition on the EXT_IRQ pin is detected while the Automatic Backup mode is on, the Security Module mode switches back to the mode it was set to before the low-level transition on EXT_IRQ. It takes up to 5 TD_SLCK cycles to complete.

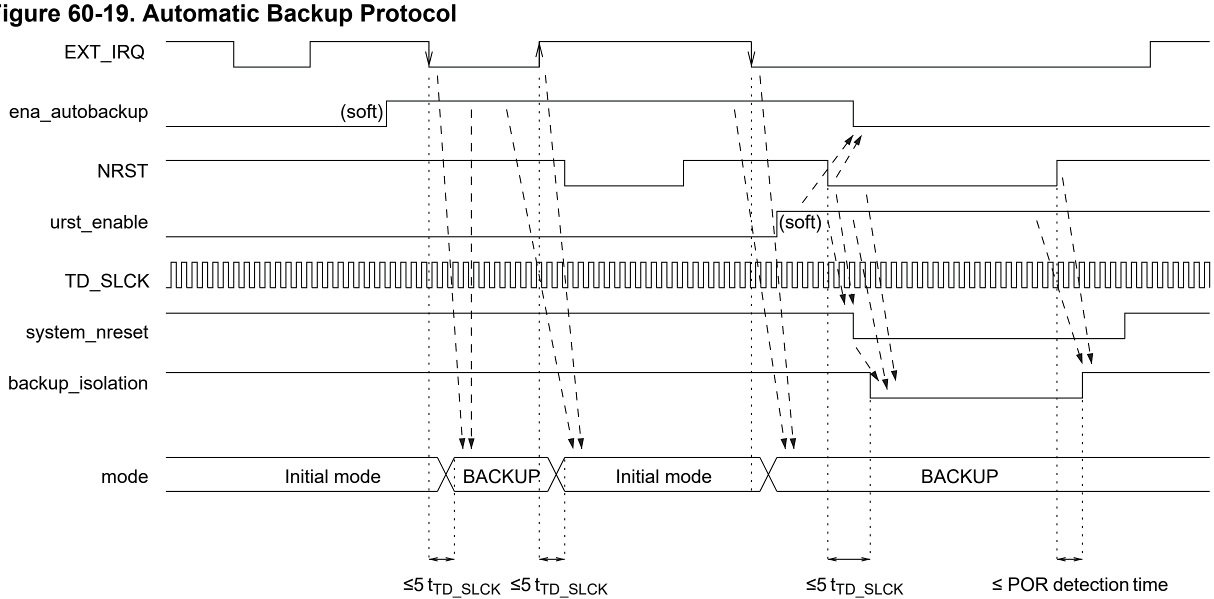
⚠WARNING The mode cannot be changed to Normal mode by software when being automatically set to Backup mode, between the EXT_IRQ low transition and the EXT_IRQ high transition, if Automatic Backup mode is on. Due to asynchronism between the system clock and TD_SLCK, that cannot be managed safely. To avoid any issue, any attempt to try and change this mode has no effect. To change the mode, first disable the Automatic Backup mode and then set the new mode manually.

If a low level is applied on the NRST pin while the Automatic Backup mode is on and the user reset is enabled in the Reset Controller, the Automatic Backup mode is reset to off in a maximum time of 5 TD_SLCK cycles.

If the Automatic Backup mode is disabled, it does not modify the current mode, but future events on the EXT_IRQ pin will have no effect as long as the Automatic Backup mode is off.

Note that EXT_IRQ is always able to generate an interrupt on the processor, whatever the value of the Automatic Backup mode. See SECUMOD_GPSBR for the selection of the external IRQ signal.

Figure 60-19. Automatic Backup Protocol



60.6.7 Activation or Deactivation of Protections

It is possible to activate or deactivate each protection separately by writing in the Normal and Backup Mode Protection registers. These registers are hidden and the only way to make them appear is to write SECUMOD_CR.KEY with the correct value. This field acts on a toggle basis: writing the correct value alternatively makes the registers appear and disappear.

At Reset state, all protections are activated except the four corresponding to the intrusion detectors (PIOBUx must be programmed).

60.6.8 Power-up Reset

After a power-up reset, the Security Module is in Backup mode, but in an unpredictable state.

The Slow Clock oscillator takes about one second to start-up. During that time, the Double Frequency Monitor can return an error to the protection manager. It is also possible that other monitors send alarms to the Protection Unit. However, a Clear command can be performed because the secure memories content is empty.

Care must be taken when writing in BUSRAM4KB or BUREG256b after reset. The user must make sure that no Erase sequence is running, otherwise the write access to BUSRAM4KB or BUREG256b is aborted. It is recommended to wait for the system to be established before accessing BUSRAM4KB or BUREG256b. This can last for at least one or two seconds. The verification is performed by reading the Status register. If there is no error for a continuous period (one second, for example), the user can access BUSRAM4KB or BUREG256b. If at least one error is detected, the user has to wait first for the ERASE_DONE flag to rise, and then wait again for at least one slow clock period after reading the Status register before writing content in the Status Clear register. At this stage, all status bits should be cleared. The user must then ensure that no error is raised in the Status register during the next second, for example.

60.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SECUMOD_CR	31:24	KEY[15:8]							
		23:16	KEY[7:0]							
		15:8						SCRAMB[1:0]		
		7:0	RESERVED	AUTOBKP[1:0]		NIMP_EN[1:0]		SWPROT	NORMAL	BACKUP
0x04	SECUMOD_SYSR	31:24								
		23:16								
		15:8								NIMP_IDLE
		7:0	SCRAMB	AUTOBKP	NIMP_EN		SWKUP	BACKUP	ERASE_ON	ERASE_DONE
0x08	SECUMOD_SR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML		REGANA	JTAG	TST	DBLFM	DWDT_SW
0x0C	SECUMOD_ASR	31:24								
		23:16								
		15:8							PSWHI	BUHI
		7:0	PSWLO	BULO	TCK	JTAG	REGANA_HI	REGANA_LO		
0x10	SECUMOD_SCR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML		REGANA	JTAG	TST	DBLFM	DWDT_SW
0x14	SECUMOD_RAMRDY	31:24								
		23:16								
		15:8								
		7:0								READY
0x18	SECUMOD_PIOBU0	31:24								
		23:16			FILTER3_5	DYNSTAT				
		15:8	SWITCH	SCHEDULE	PULLUP[1:0]			PIO_PDS	PIO_SOD	OUTPUT
		7:0	PIOBU_RFV[3:0]				PIOBU_AFV[3:0]			
0x1C	SECUMOD_PIOBU1	31:24								
		23:16			FILTER3_5	DYNSTAT				
		15:8	SWITCH	SCHEDULE	PULLUP[1:0]			PIO_PDS	PIO_SOD	OUTPUT
		7:0	PIOBU_RFV[3:0]				PIOBU_AFV[3:0]			
0x20	SECUMOD_PIOBU2	31:24								
		23:16			FILTER3_5	DYNSTAT				
		15:8	SWITCH	SCHEDULE	PULLUP[1:0]			PIO_PDS	PIO_SOD	OUTPUT
		7:0	PIOBU_RFV[3:0]				PIOBU_AFV[3:0]			
0x24	SECUMOD_PIOBU3	31:24								
		23:16			FILTER3_5	DYNSTAT				
		15:8	SWITCH	SCHEDULE	PULLUP[1:0]			PIO_PDS	PIO_SOD	OUTPUT
		7:0	PIOBU_RFV[3:0]				PIOBU_AFV[3:0]			
0x28 ... 0x57	Reserved									
0x58	SECUMOD_VBUFR	31:24								
		23:16								
		15:8								
		7:0						VBATFV[2:0]		
0x5C ... 0x63	Reserved									
0x64	SECUMOD_VCOREFR	31:24								
		23:16								
		15:8				VDDCORE_DBTV[12:8]				
		7:0	VDDCORE_DBTV[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x68	SECUMOD_VCPUFR	31:24								
		23:16								
		15:8				VDDCPU_DBTV[12:8]				
		7:0	VDDCPU_DBTV[7:0]							
0x6C ... 0x6F	Reserved									
0x70	SECUMOD_JTAGCR	31:24								
		23:16								
		15:8								
		7:0				PROC_DEBUG_MON	PROC_DEBUG_MODE[2:0]			FNTRST
0x74	SECUMOD_DYSTUNE	31:24	PERIOD[15:8]							
		23:16	PERIOD[7:0]							
		15:8	RX_OK_CORREL_NUMBER[7:0]							
		7:0	NOPA	RX_ERROR_THRESHOLD[6:0]						
0x78	SECUMOD_SCRKEY	31:24	SCRKEY[31:24]							
		23:16	SCRKEY[23:16]							
		15:8	SCRKEY[15:8]							
		7:0	SCRKEY[7:0]							
0x7C	SECUMOD_RAMACC	31:24								
		23:16								
		15:8					RW5[1:0]		RW4[1:0]	
		7:0	RW3[1:0]		RW2[1:0]		RW1[1:0]		RW0[1:0]	
0x80	SECUMOD_RAMACCSR	31:24								
		23:16								
		15:8					RW5[1:0]		RW4[1:0]	
		7:0	RW3[1:0]		RW2[1:0]		RW1[1:0]		RW0[1:0]	
0x84	SECUMOD_BMPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8					VBATH	VBATL		
		7:0	TPMH	TPML				TST	DBLFM	
0x88	SECUMOD_NMPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
0x8C	SECUMOD_NIEPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
0x90	SECUMOD_NIDPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
0x94	SECUMOD_NIMPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
		7:0	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
0x98	SECUMOD_WKPR	31:24								
		23:16			DET3	DET2	DET1	DET0		
		15:8					VBATH	VBATL		
		7:0	TPMH	TPML				TST	DBLFM	
0x9C	SECUMOD_GPSBR	31:24	KEY[11:4]							
		23:16	KEY[3:0]							
		15:8								
		7:0					EXTIRQSEL	SMCPURANGE	TSRANGE	PSWBU

60.7.1 SECUMOD Control Register

Name: SECUMOD_CR
Offset: 0x0000
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
						SCRAMB[1:0]		
Access						W	W	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	RESERVED	AUTOBKP[1:0]		NIMP_EN[1:0]		SWPROT	NORMAL	BACKUP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:16 – KEY[15:0] Password
Writing the value 0x89CA makes the Normal or Backup Protection registers appear. Writing the opposite value (0x7635) makes these registers disappear.
Writing any other value in this field has no effect.

Bits 10:9 – SCRAMB[1:0] Memory Scrambling Enable

Value	Description
0	No effect.
1	Memories are scrambled (default).
2	Memories are not scrambled.
3	No effect.

Bit 7 – RESERVED Reserved for test (write zero)

Bits 6:5 – AUTOBKP[1:0] Automatic Normal to Backup Mode Switching
Autobackup (AUTOBKP) must be enabled to configure the mode (NORMAL or BACKUP).
It is prohibited to disable autobackup (AUTOBKP) and to configure the mode (NORMAL or BACKUP) at the same time.

Value	Description
0	No effect.
1	When in Normal mode, the power-down of the core supply automatically switches to Backup mode simultaneously with core to backup isolation barrier activation (default).
2	When in Normal mode, the software must switch to Backup mode before powering down the core.
3	No effect.

Bits 4:3 – NIMP_EN[1:0] Non-Imprinting Enable

Value	Description
0	No effect.
1	The non-imprinting mechanism is authorized to start when core power is off or bus reset is asserted.

Value	Description
2	The non-imprinting mechanism is disabled (default).
3	The non-imprinting mechanism is authorized to start when core power is off or bus reset is asserted or the CPU is idle.

Bit 2 – SWPROT Software Protection

Value	Description
0	No effect.
1	Starts the BUSRAM4KB and BUREG256b Clear content.

Bit 1 – NORMAL Normal Mode

Value	Description
0	No effect.
1	Switches to Normal mode.

Bit 0 – BACKUP Backup Mode

Value	Description
0	No effect.
1	Switches to Backup mode.

60.7.2 SECUMOD System Status Register

Name: SECUMOD_SYSR
Offset: 0x0004
Reset: 0x000000D4
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								NIMP_IDLE
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	SCRAMB	AUTOBKP	NIMP_EN		SWKUP	BACKUP	ERASE_ON	ERASE_DONE
Access	R	R	R		R	R	R	R/W
Reset	1	1	0		0	1	0	0

Bit 8 – NIMP_IDLE "CPU in Idle" Preliminary Condition for Non-Imprinting

Value	Description
0	Idle is not part of the preliminary conditions list.
1	Idle is part of the preliminary conditions list.

Bit 7 – SCRAMB Scrambling Enabled

Value	Description
0	Disabled
1	Enabled

Bit 6 – AUTOBKP Automatic Backup Mode Enabled

Value	Description
0	Disabled
1	Enabled

Bit 5 – NIMP_EN Non-Imprinting Enabled

Value	Description
0	Disabled
1	Enabled

Bit 3 – SWKUP SWKUP State

Value	Description
0	No SWKUP signal sent since the last clear.
1	SWKUP signal has been sent since the last clear.

Bit 2 – BACKUP Backup Mode

Value	Description
0	Normal mode is active.

Value	Description
1	Backup mode is active.

Bit 1 – ERASE_ON Erase Process Ongoing
When ERASE_ON returns to 0, ERASE_DONE is set after half a period of ICLK.

ERASE_ON	ERASE_DONE	Status	Action
0	0	No Erase ongoing or since the last Erase.	–
1	0	An Erase process is running.	Wait until the ERASE_ON flag is reset. ERASE_DONE will rise, see row below.
0	1	An Erase occurred and is finished.	Clear the ERASE_DONE flag.
1	1	An Erase process is running. The ERASE_DONE flag refers to a previous Erase process, but was not cleared.	Wait until the ERASE_ON flag is reset, then clear the ERASE_DONE flag.

Value	Description
0	The Erase automaton is not running.
1	The Erase automaton is currently running; memories are not accessible.

Bit 0 – ERASE_DONE Erasable Memories State
Activates the SECURAM interrupt line as long as it is not cleared.

Value	Description
0	Content of the secure memories has not been erased since the last clear.
1	Content of the secure memories has been erased since the last clear. Write a '1' to this bit to clear this flag. The flag also activates the SECURAM interrupt line as long as it is not cleared. **Note:** Not clearing this flag does not prevent the next erase processes.

60.7.3 SECUMOD Status Register

Name: SECUMOD_SR
Offset: 0x0008
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No alarm generated since the last clear.

1: An alarm has been generated by the corresponding monitor since the last clear.

Note: Even unprotected detectors, such as those without a corresponding bit set in SECUMOD_NMPR or SECUMOD_BMPR, can set a flag in SECUMOD_SR, but no erase is performed.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			R	R	R	R		
Reset			0	0	0	0		

Bit	15	14	13	12	11	10	9	8
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
	TPMH	TPML		REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	R	R		R	R	R	R	R
Reset	0	0		0	0	0	0	0

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor

Bit 11 – VBATH High VBAT Voltage Monitor

Bit 10 – VBATL Low VBAT Voltage Monitor

Bit 7 – TPMH High Temperature Monitor

Bit 6 – TPML Low Temperature Monitor

Bit 4 – REGANA VDDANA Regulator Monitor
See SECUMOD_ASR for more information.

Bit 3 – JTAG JTAG Pins Monitor

Bit 2 – TST Test Pin Monitor

Bit 1 – DBLFM Double Frequency Monitor

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm

60.7.4 SECUMOD Auxiliary Status Register

Name: SECUMOD_ASR
Offset: 0x000C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No alarm generated since the last clear.

1: An alarm has been generated by the corresponding monitor since the last clear.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							PSWHI	BUHI
Access							R	R
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	PSWLO	BULO	TCK	JTAG	REGANA_HI	REGANA_LO		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bit 9 – PSWHI VDDIN33 (used as secondary LDO power source through backup power switch) low alarm detected is the cause of VBATL flag in SECUMOD_SR

Bit 8 – BUHI VBAT high alarm detected is the cause of VBATH flag in SECUMOD_SR

Bit 7 – PSWLO VDDIN33 (used as secondary LDO power source through backup power switch) low alarm detected is the cause of VBATL flag in SECUMOD_SR

Bit 6 – BULO VBAT low alarm detected is the cause of VBATL flag in SECUMOD_SR

Bit 5 – TCK TCK/TMS activity detected is the cause of JTAG flag in SECUMOD_SR

Bit 4 – JTAG JTAGSEL or processor debug acknowledge is the cause of JTAG flag in SECUMOD_SR

Bit 3 – REGANA_HI High voltage alarm from VDDANA regulator is the cause of REGANA flag in SECUMOD_SR

Bit 2 – REGANA_LO Low voltage alarm from VDDANA regulator is the cause of REGANA flag in SECUMOD_SR

60.7.5 SECUMOD Status Clear Register

Name: SECUMOD_SCR
Offset: 0x0010
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Clears the corresponding alarm flag bit.

If the corresponding alarm was programmed to generate an SWKUP signal, clearing the alarm also clears SECUMOD_SYSR.SWKUP.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			W	W	W	W		
Reset			–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TPMH	TPML		REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	R	R		W	W	W	W	W
Reset	–	–		–	–	–	–	–

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor

Bit 11 – VBATH High VBAT Voltage Monitor

Bit 10 – VBATL Low VBAT Voltage Monitor

Bit 7 – TPMH High Temperature Monitor

Bit 6 – TPML Low Temperature Monitor

Bit 4 – REGANA VDDANA Regulator Monitor

Bit 3 – JTAG JTAG Pins Monitor

Bit 2 – TST Test Pin Monitor

Bit 1 – DBLFM Double Frequency Monitor

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm

60.7.6 SECUMOD RAM Access Ready Register

Name: SECUMOD_RAMRDY
Offset: 0x0014
Reset: undefined
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								READY
Access								R
Reset								

Bit 0 – READY Ready for System Access Flag
When exiting a CPU idle, a system reset or Backup mode, this flag must be read high before accessing the secure memories. The flag remains low until any ongoing process stops.

60.7.7 SECUMOD PIO Backup Register x

Name: SECUMOD_PIOBUx
Offset: 0x18 + x*0x04 [x=0..3]
Reset: 0x00001000
Property: Read/Write

The FILTER3_5 and DYNSTAT fields only exist for PIOBUs with an even index.

Important: In order to cover internal resynchronization times, at least 30 µs must elapse between two write accesses to a SECUMOD_PIOBU register when modifying fields related to dynamic or static intrusion settings. However, modifications on IO direction, pull-up/pull-down or level do not require such precautions.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
			FILTER3_5	DYNSTAT				
Access			R/W	R/W				
Reset			0	0				
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	SWITCH	SCHEDULE	PULLUP[1:0]			PIO_PDS	PIO_SOD	OUTPUT
Access	R/W	R/W	R/W	R/W		RO	R/W	R/W
Reset	0	0	0	1		0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PIOBU_RFV[3:0]				PIOBU_AFV[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 21 – FILTER3_5 Filter for Dynamic Signatures Input

Value	Description
0	3-stage majority vote (default).
1	5-stage majority vote.

Bit 20 – DYNSTAT Switch for Static or Dynamic Detection Intrusion

Important:
When the Dynamic Intrusion mode is selected, the user must write 0 to the PIOBU_AFV and PIOBU_RFV fields and write a ‘1’ to OUTPUT for the two PIOs of the dynamic pair, to deactivate Static Detection. If one of these fields is not completely reset, the Static Detection mechanism is still partially running and can interpret the dynamic signal as an attack. The two detection modes are incompatible on the same pins.

If the application requires dynamic signatures to be stopped/restarted several times, it is recommended to set the detection threshold to a value greater or equal to 2 in SECUMOD_DYSTUNE in RX_ERROR_THRESHOLD, or to mask the PIO protection during the first millisecond following each dynamic signature start.

Value	Description
0	Static detection intrusion (default).

Value	Description
1	Dynamic detection intrusion.

Bit 15 – SWITCH Switch State for Intrusion Detection

Value	Description
0	Input default state is low level.
1	Input default state is high level.

Bit 14 – SCHEDULE Pull-up/Pull-down Scheduled

Value	Description
0	Pull-up/Pull-down is not scheduled.
1	Pull-up/Pull-down is scheduled.

Bits 13:12 – PULLUP[1:0] Programmable Pull-up State
Used to control the internal pull-up or pull-down.

PULLUP		Description
0	0	No pull-up / pull-down connected.
0	1	Pull-up connected.
1	0	Pull-down connected.
1	1	Reserved

Bit 10 – PIO_PDS Level on the Pin in Input Mode (OUTPUT = 0)

Value	Description
0	The I/O line is at level 0.
1	The I/O line is at level 1.

Bit 9 – PIO_SOD Set/Clear the I/O Line when configured in Output Mode (OUTPUT =1)

Value	Description
0	Clears the data to be driven on the I/O line.
1	Sets the data to be driven on the I/O line.

Bit 8 – OUTPUT Configure I/O Line in Input/Output

Value	Description
0	The I/O line is a pure input.
1	The I/O line is enabled in output.

Bits 7:4 – PIOBU_RFV[3:0] PIOBUx Reset Filter Value
Defines the number of consecutive valid states to be reached before resetting the AFV counter.
Must be set to 0 when Dynamic Intrusion is selected.

PIOBU_RFV	Maximum Counter Value
0	0 (No static protection)
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512

Bits 3:0 – PIOBU_AFV[3:0] PIOBU Alarm Filter Value
Used to define the filter value prior to generating an alarm.
Must be set to 0 when Dynamic Intrusion is selected.

PIOBU_AFV	Maximum Counter Value
0	0 (No static protection)
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512

60.7.8 SECUMOD VBAT Filter Register

Name: SECUMOD_VBUFR
Offset: 0x0058
Reset: 0x00000007
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						VBATFV[2:0]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bits 2:0 – VBATFV[2:0] VBAT Filter Value
0 to 7: This field is used to define the filter value for the VBAT and VDDBU voltage monitor.

60.7.9 SECUMOD VDDCORE Filter Register

Name: SECUMOD_VCOREFR
Offset: 0x0064
Reset: 0x00001FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				VDDCORE_DBTV[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	VDDCORE_DBTV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 12:0 – VDDCORE_DBTV[12:0] VDDCORE Programmable Debouncing Time Value
This field is used to define the debouncing filter value.
The debouncing time is computed as follows:
Debouncing Time = VDDCORE_DBTV/f_{TD_SCLK}.
At reset, VDDCPU_DBTV is set to the maximum value, 8191, corresponding to a typical debouncing value of 250 ms.

60.7.10 SECUMOD VDDCPU Filter Register

Name: SECUMOD_VCPUFR
Offset: 0x0068
Reset: 0x00001FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				VDDCPU_DBTV[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	VDDCPU_DBTV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 12:0 – VDDCPU_DBTV[12:0] VDDCPU Programmable Debouncing Time Value
This field is used to define the debouncing filter value.
The debouncing time is computed as follows:
Debouncing Time = VDDCPU_DBTV/f_{TD_SCLK}
At reset, VDDCPU_DBTV is set to the maximum value, 8191, corresponding to a typical debouncing value of 250 ms.

60.7.11 SECUMOD JTAG Protection Control Register

Name: SECUMOD_JTAGCR
Offset: 0x0070
Reset: see Note
Property: Read/Write

Note: The register reset value is 0x00000000 when the JTAG DEFDBG bit field in the OTP JTAG DISABLE packet is not programmed.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				PROC_DEBUG_MON	PROC_DEBUG_MODE[2:0]			FNTRST
Access				R/W	R/W	R/W	R/W	R/W
Reset								

Bit 4 – PROC_DEBUG_MON Debug Acknowledge (DBGACK) Monitoring

Value	Description
0	The Arm processor pin DBGACK is not monitored; as a consequence, the software can access debug features of the processor without causing an intrusion in the Security Module.
1	The Arm processor pin DBGACK is monitored. Processor entering in Debug mode triggers an intrusion.

Bits 3:1 – PROC_DEBUG_MODE[2:0] Invasive/Non-Invasive Secure/Non-Secure Debug Permissions
This field is used to set different debug permission levels. For instance, it can be used to prevent debug on secure parts of the code. The table below shows the effect of the field value on the processor pins (SPIDEN, DBGEN, SPNIDEN and NIDEN).

PROC_DEBUG_MODE Value	Debug Permissions	SPIDEN	DBGEN	SPNIDEN	NIDEN
b000	No Debug	0	0	0	0
b001	Non-Invasive, Non-Secure	0	0	0	1
b010	Full Non-Secure (Invasive and Non-Invasive)	0	1	0	0
b011	Full Non-Secure + Non-Invasive Secure	0	1	1	1
b100	Full Debug allowed	1	1	1	1

Bit 0 – FNTRST Force NTRST

Value	Description
0	The Arm processor TAP controller access and Boundary JTAG are not blocked by the Security Module.
1	nDBGRESET of the Arm processor TAP controller and Boundary JTAG reset are held low, preventing the processor to switch to Debug state and Boundary JTAG to work.

60.7.12 SECUMOD Dynamic Signatures Tuning Register

Name: SECUMOD_DYSTUNE
Offset: 0x0074
Reset: 0x00010501
Property: Read/Write

It is recommended to disable the protection on dynamic intrusions before modifying this register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	RX_OK_CORREL_NUMBER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	0	1

Bit	7	6	5	4	3	2	1	0
	NOPA	RX_ERROR_THRESHOLD[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:16 – PERIOD[15:0] Signature Clock Period
This 32-bit field contains the dividing ratio applied on ICLK to build the signature clock:
t_{SIGCLK} =(8*PERIOD) * t_{ICLK}.
Writing 0 has no effect: 1 ≤ PERIOD ≤ 65535
Note: Patterns generated by dynamic signatures have a mean length of 4 cycles of the signature clock.

Bits 15:8 – RX_OK_CORREL_NUMBER[7:0] Error Counter Reset Threshold
This 16-bit field contains the number of consecutive matching patterns which must be received to consider that the external signature carrier is safe (again), and to forget any ongoing dynamic intrusion.
Writing values lower than 0x5 has no effect: 5 ≤ RX_OK_CORREL_NUMBER ≤ 255

Bit 7 – NOPA No Periodic Alarm

Value	Description
0	The alarm is regenerated periodically while intrusion is maintained.
1	The alarm is not regenerated periodically while intrusion is maintained.

Bits 6:0 – RX_ERROR_THRESHOLD[6:0] Error Detection Threshold
This 7-bit field contains the number of mismatching patterns which must be received to trigger an alarm.
Writing 0 has no effect: 1 ≤ RX_ERROR_THRESHOLD ≤ 127

60.7.13 SECUMOD Scrambling Key Register

Name: SECUMOD_SCRKEY
Offset: 0x0078
Reset: undefined
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SCRKEY[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	23	22	21	20	19	18	17	16
	SCRKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	15	14	13	12	11	10	9	8
	SCRKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								
Bit	7	6	5	4	3	2	1	0
	SCRKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 31:0 – SCRKEY[31:0] Scrambling Key Value
This 32-bit key is used by the secure memories scrambler/descrambler logic. When changed, the readable content of the memories is made unintelligible instantaneously.

60.7.14 SECUMOD RAM Access Rights Register

Name: SECUMOD_RAMACC
Offset: 0x007C
Reset: 0x00000FFF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

00: No access allowed

01: Only write access allowed

10: Only read access allowed

11: Read and write accesses allowed

Accessing a forbidden area causes an interrupt (SECURAM ID).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RW5[1:0]		RW4[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	RW3[1:0]		RW2[1:0]		RW1[1:0]		RW0[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 11:10 – RW5[1:0] Access right for RAM region [5 Kbytes; 6 Kbytes] (register bank BUREG256b)

Bits 9:8 – RW4[1:0] Access right for RAM region [4 Kbytes; 5 Kbytes]

Bits 7:6 – RW3[1:0] Access right for RAM region [3 Kbytes; 4 Kbytes]

Bits 5:4 – RW2[1:0] Access right for RAM region [2 Kbytes; 3 Kbytes]

Bits 3:2 – RW1[1:0] Access right for RAM region [1 Kbyte; 2 Kbytes]

Bits 1:0 – RW0[1:0] Access right for RAM region [0; 1 Kbyte]

60.7.15 SECUMOD RAM Access Rights Status Register

Name: SECUMOD_RAMACCSR
Offset: 0x0080
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

00: No access violation occurred

01: Write access violation occurred

10: Read access violation occurred

11: Read and write access violation occurred

Writing any value to this register resets the register and the associated interrupt line (SECURAM ID).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RW5[1:0]		RW4[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RW3[1:0]		RW2[1:0]		RW1[1:0]		RW0[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:10 – RW5[1:0] Access right status for RAM region [5 Kbytes; 6 Kbytes] (register bank BUREG256b)

Bits 9:8 – RW4[1:0] Access right status for RAM region [4 Kbytes; 5 Kbytes]

Bits 7:6 – RW3[1:0] Access right status for RAM region [3 Kbytes; 4 Kbytes]

Bits 5:4 – RW2[1:0] Access right status for RAM region [2 Kbytes; 3 Kbytes]

Bits 3:2 – RW1[1:0] Access right status for RAM region [1 Kbytes; 2 Kbytes]

Bits 1:0 – RW0[1:0] Access right status for RAM region [0; 1 Kbyte]

60.7.16 SECUMOD Backup Mode Protection Register

Name: SECUMOD_BMPR
Offset: 0x0084
Reset: 0x000F0CC7
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Protection disabled.

1: Protection enabled.

Remember: Enabling PIOBU protection requires additional programming of PIOBUx registers.

Note: PIO backup protections are off after backup reset whatever the reset value of this register. See SECUMOD_PIOBUx to enable these protections.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			R/W	R/W	R/W	R/W		
Reset			0	0	1	1		

Bit	15	14	13	12	11	10	9	8
					VBATH	VBATL		
Access					R/W	R/W		
Reset					1	1		

Bit	7	6	5	4	3	2	1	0
	TPMH	TPML				TST	DBLFM	
Access	R/W	R/W				R/W	R/W	
Reset	1	1				1	1	

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection

Bit 11 – VBATH High VBAT Voltage Monitor Protection

Bit 10 – VBATL Low VBAT Voltage Monitor Protection

Bit 7 – TPMH High Temperature Monitor Protection

Bit 6 – TPML Low Temperature Monitor Protection

Bit 2 – TST Test Pin Monitor Protection

Bit 1 – DBLFM Double Frequency Monitor Protection

60.7.17 SECUMOD Normal Mode Protection Register

Name: SECUMOD_NMPR
Offset: 0x0088
Reset: 0x003CFCFF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Protection disabled.

1: Protection enabled.

Remember: Enabling PIOBU protection requires additional programming of PIOBUx registers.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			R/W	R/W	R/W	R/W		
Reset			1	1	1	1		

Bit	15	14	13	12	11	10	9	8
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	1	1	1	1	1	1		

Bit	7	6	5	4	3	2	1	0
	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor Protection

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor Protection

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor Protection

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor Protection

Bit 11 – VBATH High VBAT Voltage Monitor Protection

Bit 10 – VBATL Low VBAT Voltage Monitor Protection

Bit 7 – TPMH High Temperature Monitor Protection

Bit 6 – TPML Low Temperature Monitor Protection

Bit 5 – MBZ Must be set to zero

Bit 4 – REGANA VDDANA Regulator Monitor Protection

Bit 3 – JTAG JTAG Pin Protection

Bit 2 – TST Test Pin Protection

Bit 1 – DBLFM Double Frequency Monitor Protection

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm Protection

60.7.18 SECUMOD Normal Interrupt Enable Protection Register

Name: SECUMOD_NIEPR
Offset: 0x008C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
			DET3	DET2	DET1	DET0		
Access			R/W	R/W	R/W	R/W		
Reset			–	–	–	–		
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection Interrupt Enable

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor Protection Interrupt Enable

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor Protection Interrupt Enable

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor Protection Interrupt Enable

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor Protection Interrupt Enable

Bit 11 – VBATH High VBAT Voltage Monitor Protection Interrupt Enable

Bit 10 – VBATL Low VBAT Voltage Monitor Protection Interrupt Enable

Bit 7 – TPMH High Temperature Monitor Protection Interrupt Enable

Bit 6 – TPML Low Temperature Monitor Protection Interrupt Enable

Bit 5 – MBZ Must be set to zero

Bit 4 – REGANA VDDANA Regulator Monitor Protection Interrupt Enable

Bit 3 – JTAG JTAG Pin Protection Interrupt Enable

Bit 2 – TST Test Pin Protection Interrupt Enable

Bit 1 – DBLFM Double Frequency Monitor Protection Interrupt Enable

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm Protection Interrupt Enable

60.7.19 SECUMOD Normal Interrupt Disable Protection Register

Name: SECUMOD_NIDPR
Offset: 0x0090
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			W	W	W	W		
Reset			–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection Interrupt Disable

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor Protection Interrupt Disable

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor Protection Interrupt Disable

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor Protection Interrupt Disable

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor Protection Interrupt Disable

Bit 11 – VBATH High VBAT Voltage Monitor Protection Interrupt Disable

Bit 10 – VBATL Low VBAT Voltage Monitor Protection Interrupt Disable

Bit 7 – TPMH High Temperature Monitor Protection Interrupt Disable

Bit 6 – TPML Low Temperature Monitor Protection Interrupt Disable

Bit 5 – MBZ Must be set to zero

Bit 4 – REGANA VDDANA Regulator Monitor Protection Interrupt Disable

Bit 3 – JTAG JTAG Pin Protection Interrupt Disable

Bit 2 – TST Test Pin Protection Interrupt Disable

Bit 1 – DBLFM Double Frequency Monitor Protection Interrupt Disable

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm Interrupt Disable

60.7.20 SECUMOD Normal Interrupt Mask Protection Register

Name: SECUMOD_NIMPR
Offset: 0x0094
Reset: see Note
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Note: Register reset value is 0x00000000 after peripheral reset. Other reset values are defined after backup reset.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
			DET3	DET2	DET1	DET0		
Access			R	R	R	R		
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	VDDCPUH	VDDCOREH	VDDCPUL	VDDCOREL	VBATH	VBATL		
Access	R	R	R	R	R	R		
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TPMH	TPML	MBZ	REGANA	JTAG	TST	DBLFM	DWDT_SW
Access	R	R	R	R	R	R	R	R
Reset								

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection Interrupt Mask

Bit 15 – VDDCPUH High VDDCPU Voltage Monitor Protection Interrupt Mask

Bit 14 – VDDCOREH High VDDCORE Voltage Monitor Protection Interrupt Mask

Bit 13 – VDDCPUL Low VDDCPU Voltage Monitor Protection Interrupt Mask

Bit 12 – VDDCOREL Low VDDCORE Voltage Monitor Protection Interrupt Mask

Bit 11 – VBATH High VBAT Voltage Monitor Protection Interrupt Mask

Bit 10 – VBATL Low VBAT Voltage Monitor Protection Interrupt Mask

Bit 7 – TPMH High Temperature Monitor Protection Interrupt Mask

Bit 6 – TPML Low Temperature Monitor Protection Interrupt Mask

Bit 5 – MBZ Must be set to zero

Bit 4 – REGANA VDDANA Regulator Monitor Protection Interrupt Mask

Bit 3 – JTAG JTAG Pin Protection Interrupt Mask

Bit 2 – TST Test Pin Protection Interrupt Mask

Bit 1 – DBLFM Double Frequency Monitor Protection Interrupt Mask

Bit 0 – DWDT_SW Programmable Secure Watchdog Alarm Interrupt Mask

60.7.21 SECUMOD Wake-up Register

Name: SECUMOD_WKPR
Offset: 0x0098
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: No wake-up signal is generated if the corresponding alarm is detected.

1: A wake-up signal (SWKUP) is generated if the corresponding alarm is detected.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			DET3	DET2	DET1	DET0		
Access			R/W	R/W	R/W	R/W		
Reset			0	0	0	0		
Bit	15	14	13	12	11	10	9	8
					VBATH	VBATL		
Access					R/W	R/W		
Reset					0	0		
Bit	7	6	5	4	3	2	1	0
	TPMH	TPML				TST	DBLFM	
Access	R/W	R/W				R/W	R/W	
Reset	0	0				0	0	

Bits 18, 19, 20, 21 – DETx PIOBU Intrusion Detector Protection

Bit 11 – VBATH High VBAT Voltage Monitor Protection

Bit 10 – VBATL Low VBAT Voltage Monitor Protection

Bit 7 – TPMH High Temperature Monitor Protection

Bit 6 – TPML Low Temperature Monitor Protection

Bit 2 – TST Test Pin Monitor Protection

Bit 1 – DBLFM Double Frequency Monitor Protection

60.7.22 SECUMOD General Purpose Security Bits Register

Name: SECUMOD_GPSBR
Offset: 0x009C
Reset: 0x00000111
Property: Read/Write

This register enables the control of environmental sensors or security mechanisms linked to the Security Module.

Bit	31	30	29	28	27	26	25	24
	KEY[11:4]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	KEY[3:0]							
Access	W	W	W	W				
Reset	0	0	0	0				
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					EXTIRQSEL	SMCPURANGE	TSRANGE	PSWBU
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bits 31:20 – KEY[11:0] Safety key. Must write 0xD5E to enable GPSBR modifications

Bit 3 – EXTIRQSEL

Value	Description
0	EXT_IRQ0
1	EXT_IRQ1

Bit 2 – SMCPURANGE

Value	Description
0	Adjusts SM VDDCPU thresholds for 600 MHz max frequency operation.
1	Adjusts SM VDDCPU thresholds for 800 MHz max frequency operation.

Bit 1 – TSRANGE

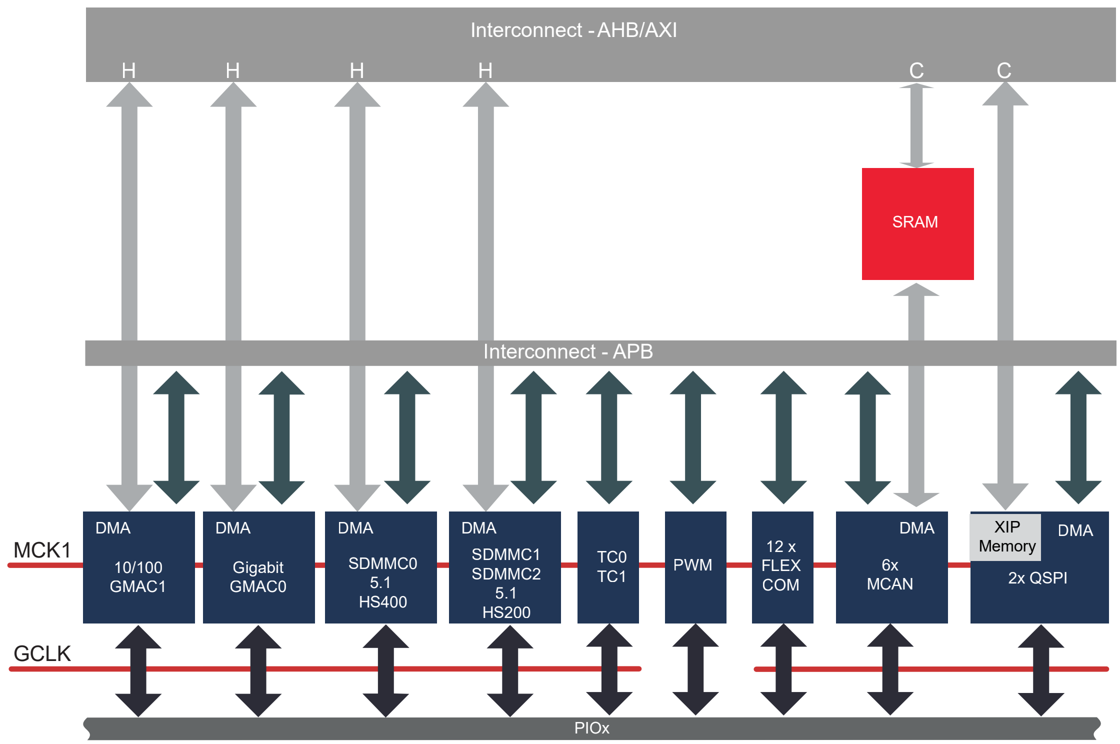
Value	Description
0	Sets temperature sensor high threshold to 105°C.
1	Sets temperature sensor high threshold to 120°C.

Bit 0 – PSWBU Power Source Switch Backup Domain
Enables the backup domain power source to automatically switch from VDDIN33 to VBAT supply when the SECUMOD enters Backup mode. Refer to the section "Special Function Register Backup (SFRBU)" for details on backup power switch control.

61. CONNECTIVITY SUBSYSTEM

61.1 Block Diagram

Figure 61-1. Connectivity Subsystem Block Diagram



61.2 Components

- 10/100/1000 Ethernet MAC (GMAC0) and 10/100 Ethernet MAC (GMAC1)
- 3x Secure Digital MultiMedia Card Controller (SDMMC0, 1, 2)
- Timer Counter (TC0, 1)
- Pulse Width Modulation Controller (PWM)
- 12x Flexible Serial Communication Controller (FLEXCOM)
- 6x Controller Area Network (MCAN)
- 1x Octal Serial Peripheral Interface and 1x Quad Serial Peripheral Interface (QSPI)

61.3 FLEXCOM Features

FLEXCOM	Subfunction	0	1	2	3	4	5	6	7	8	9	10	11
TWI	Normal/Fast (Client/Host)/FM+	x	x	x	x	x	x	x	x	x	x	x	x
	Alternate command	x	x	x	x	x	x	x	x	x	x	x	x
	3 Client ADDR	x	x	x	x	x	x	x	x	x	x	x	x
	High-speed	x	x	x	x	x	x	x	x	x	x	x	x
	SMBUS with Alert pin	x	x	x	x	x	x	x	x	x	x	x	x
	Sniffer	x	x	x	x	x	x	x	x	x	x	x	x
	Asynchronous partial wake-up	x	x	x	x	x	x	x	x	x	x	x	x
	FIFO size: 32 words	x	x	x	x	x	x	x	x	x	x	x	x
USART	Basic	x	x	x	x	x	x	x	x	x	x	x	x
	Hardware handshaking/RS485	x	x	x	x	x	x	x	x	x	x	x	x
	ISO7816	x	x	x	x	x	x	x	x	x	x	x	x
	LIN	x	x	x	x	x	x	x	x	x	x	x	x
	IrDA	x	x	x	x	x	x	x	x	x	x	x	x
	Manchester	x	x	x	x	x	x	x	x	x	x	x	x
	Asynchronous partial wake-up	x	x	x	x	x	x	x	x	x	x	x	x
	FIFO size: 32 words	x	x	x	x	x	x	x	x	x	x	x	x
SPI	4 CS	x	–	–	x	–	–	x	–	–	x	–	–
	Asynchronous partial wake-up	x	x	x	x	x	x	x	x	x	x	x	x
	FIFO size: 32 words	x	x	x	x	x	x	x	x	x	x	x	x

61.4 Product Dependencies

61.4.1 Clocks

All the components of this subsystem have their clocks controlled by the PMC, which is a part of the system controller.

The connectivity peripherals are part of the APS and HSS matrixes, and are therefore clocked by MCK1. All, except the PWM controller, have a GLCK input to generate fixed baud rates, independent from the CPU frequency.

61.4.1.1 QSPI Features

The QSPI Core peripheral is fed with the peripheral clock or QSPI GCLK.

QSPI0 embeds a DLL, enabling QSPI0 to reach 100 MHz on data with a 100-MHz GCLK clock.

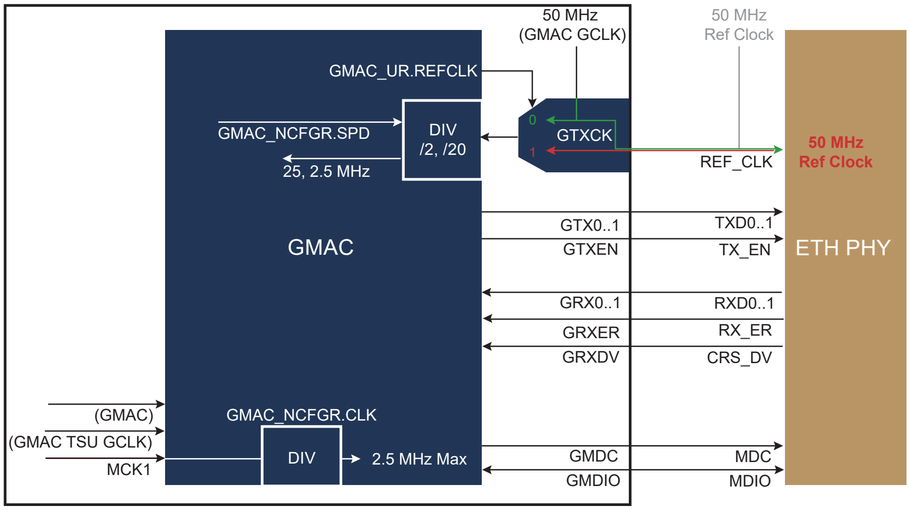
QSPI1 needs a 150-MHz GCLK to reach 75 MHz on data.

Refer to Electrical Characteristics for timing extractions in different modes.

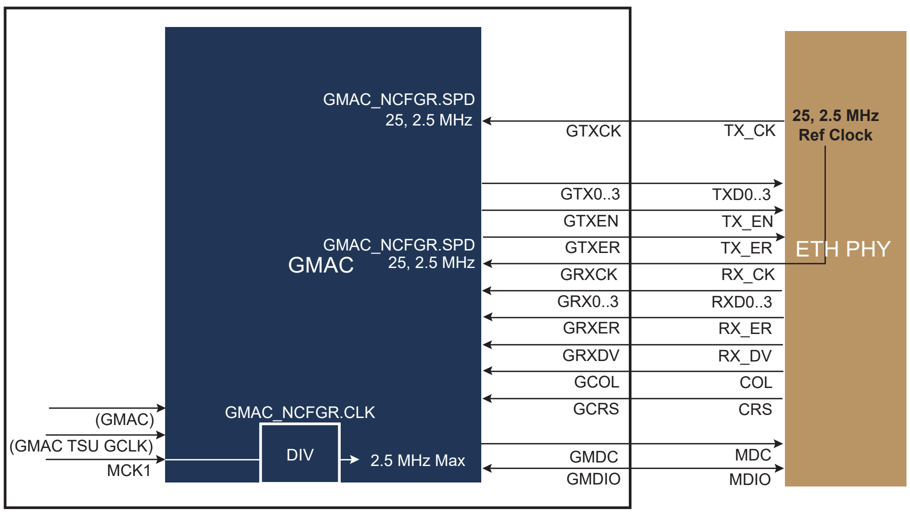
61.4.1.2 GMAC Features

- GMAC0 supports RGMII, MII, RMII
- GMAC1 supports MII, RMII
- The media interface is configured using the GMAC_UR.MIM and GMAC_NCR.MII bits. Each interface has a dedicated clock configuration.

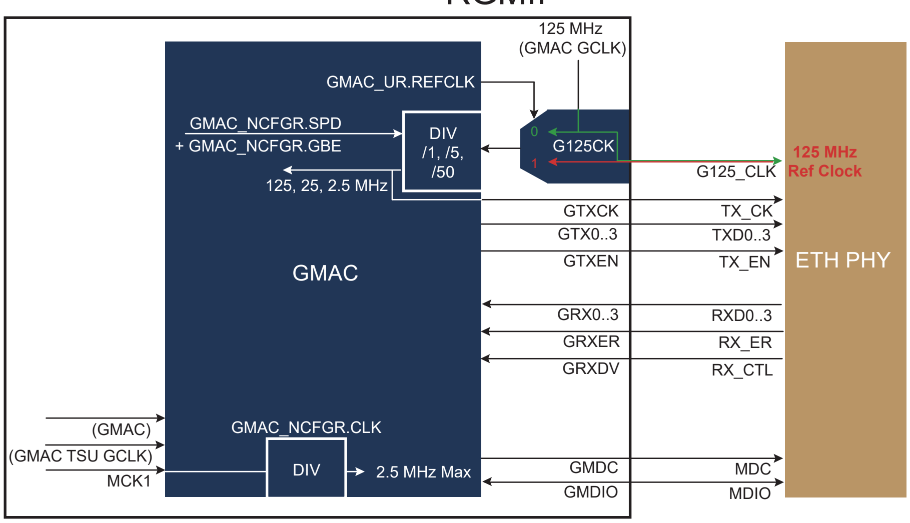
RMII



MII



RGMII



61.4.2 Interrupts

Refer to the table Peripheral Identifiers.

61.4.3 Reset

Connectivity peripherals are connected to the processor and peripherals reset line.

61.4.4 I/Os

I/Os are multiplexed on PIOs.

For the applicable I/O type (General Purpose (GPIO) or High-Speed (HSIO)) and power supply, refer to the table Pin Description. I/O drive and slew rate configuration differs depending on the I/O type. Refer to I/O characteristics in the section Electrical Characteristics.

61.4.4.1 GMAC I/Os

For RGMII operations, G0_TXCK (PA24 PIO line) must be configured with the internal pull-up resistor enabled to avoid a floating line when connecting an Ethernet PHY having a high input impedance.

The device GMAC interface complies with the RGMII v1.3 specification that requires on the TX side a typical 2 ns data-to-clock delay at PCB level between the transmitting port (device) and the receiving port (Ethernet PHY). In practice, it is convenient to select Ethernet PHYs that can add this 2 ns delay on their TXC input. As an example, Microchip KSZ9131 features an internal DLL to delay the TXC line.

61.4.4.2 QSPI I/Os

QSPI_IO2 and QSPI_IO3 are dual functions and used as HOLD and WP in Dual and Serial I/O mode. As Quad mode is not the default mode, they need to be pulled up until Quad mode is enabled in the Configuration register. Pull-up resistors are not required for QSPI_IO0 and QSPI_IO1.

61.5 Special Functions in SFR/SFRBU

None.

62. Gigabit Ethernet MAC (GMAC)

62.1 Description

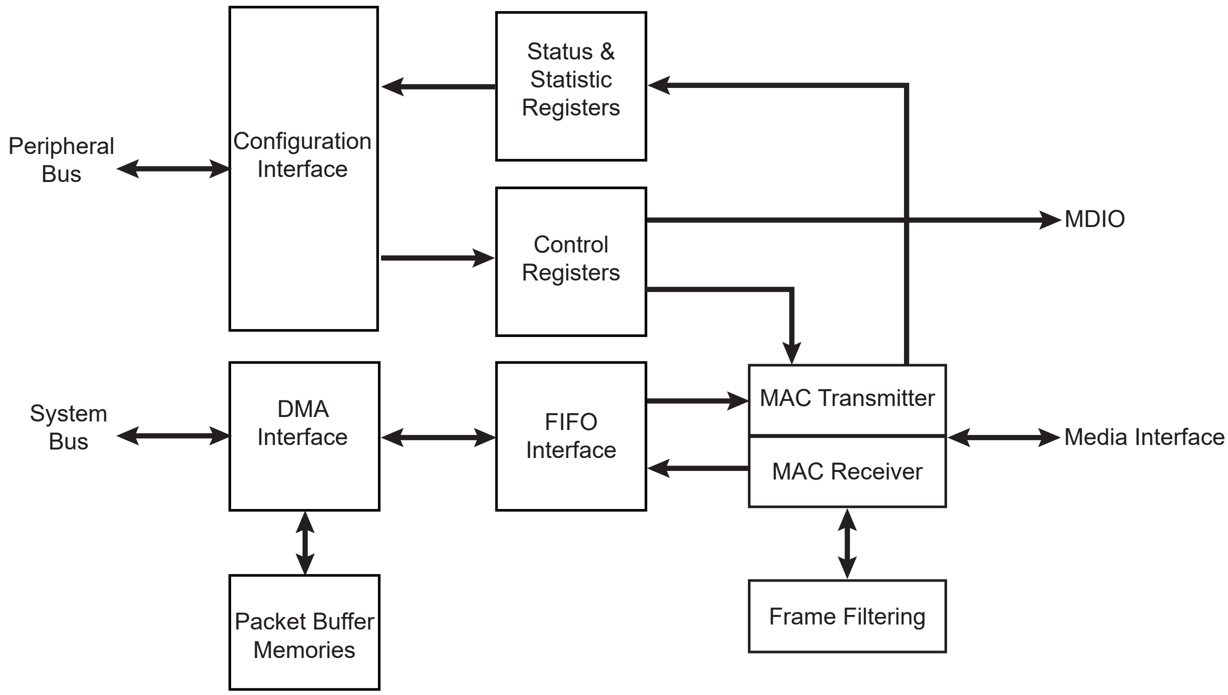
The Gigabit Ethernet MAC (GMAC) module implements a 10/100/1000 Mbps Ethernet MAC compatible with the IEEE 802.3 standard. The GMAC can operate in either half or full duplex mode at all supported speeds. The GMAC Network Configuration Register is used to select the speed, duplex mode and interface type (MII, RMII, RGMII).

62.2 Embedded Characteristics

- Compatible with IEEE Standard 802.3
- 10, 100 and 1000 Mbps Operation
- Supports 802.1Qav Traffic Shaping on Two Highest Priority Queues
- Full and Half Duplex Operation at all Supported Speeds of Operation
- Statistics Counter Registers for RMON/MIB
- MII/RMII/RGMII Interface to the Physical Layer
- Integrated Physical Coding
- Direct Memory Access (DMA) Interface to External Memory
- Support for 6 Priority Queues
- 24 Kbytes Transmit Local Memory and 8 Kbytes Receive Local Memory (refer to Table 62-4 for queue-specific sizes)
- Programmable Burst Length and Endianism for DMA
- Interrupt Generation to Signal Receive and Transmit Completion, Errors or Other Events
- Automatic Pad and Cyclic Redundancy Check (CRC) Generation on Transmitted Frames
- Frame Extension and Frame Bursting at 1000 Mbps in Half Duplex Mode
- Automatic Discard of Frames Received with Errors
- Receive and Transmit IP, TCP and UDP Checksum Offload. Both IPv4 and IPv6 Packet Types Supported
- Address Checking Logic for Four Specific 48-bit Addresses, Four Type IDs, Promiscuous Mode, Hash Matching of Unicast and Multicast Destination Addresses and Wake-on-LAN
- Management Data Input/Output (MDIO) Interface for Physical Layer Management
- Support for Jumbo Frames up to 10240 Bytes
- Full Duplex Flow Control with Recognition of Incoming Pause Frames and Hardware Generation of Transmitted Pause Frames
- Half Duplex Flow Control by Forcing Collisions on Incoming Frames
- Support for 802.1Q VLAN Tagging with Recognition of Incoming VLAN and Priority Tagged Frames
- Support for 802.1Qbb Priority-based Flow Control
- Programmable Inter Packet Gap (IPG) Stretch
- Recognition of IEEE 1588 PTP Frames
- IEEE 1588 Timestamp Unit (TSU)
- Support for 802.1AS Timing and Synchronization

62.3 Block Diagram

Figure 62-1. Block Diagram



62.4 Signal Interfaces

The GMAC includes the following signal interfaces:

- Media interface supports MII/RMII/RGMII and connects to the external PHY
- Management Data Input/Output (MDIO) connects to the external PHY for management
- Configuration interface
- System bus interface for direct memory access (DMA)
- GTSUCOMP signal for TSU timer count value comparison

Table 62-1. GMAC Connections to PHY in Different Modes

Signal Name	Function	MII	RMII	RGMII
GTXCK	Transmit Clock or Reference Clock	TXCK	REFCK	TXCK
G125CK	125 MHz input Clock	Not Used	Not Used	125 MHz Ref Clk
GTXEN	Transmit Enable	TXEN	TXEN	TXCTL
GTX[3..0]	Transmit Data	TXD[3:0]	TXD[1:0]	TXD[3:0]
GTXER	Transmit Coding Error	TXER	Not Used	Not Used
GRXCK	Receive Clock	RXCK	Not Used	RXCK
GRXDV	Receive Data Valid	RXDV	CRSDV	RXCTL
GRX[3..0]	Receive Data	RXD[3:0]	RXD[1:0]	RXD[3:0]

...........continued

Signal Name	Function	MII	RMII	RGMII
GRXER	Receive Error	RXER	RXER	Not Used
GCRS	Carrier Sense and Data Valid	CRS	Not Used	Not Used
GCOL	Collision Detect	COL	Not Used	Not Used
GMDC	Management Data Clock	MDC	MDC	MDC
GMDIO	Management Data Input/Output	MDIO	MDIO	MDIO

62.5 Product Dependencies

62.5.1 I/O Lines

The pins used for interfacing the GMAC may be multiplexed with PIO lines. The programmer must first program the PIO Controller to assign the pins to their peripheral function. If I/O lines of the GMAC are not used by the application, they can be used for other purposes by the PIO Controller.

62.5.2 Power Management

The GMAC is not continuously clocked. The user must first enable the GMAC clock in the Power Management Controller before using it.

62.5.3 Interrupt Sources

The GMAC interrupt line is connected to one of the internal sources of the interrupt controller. Using the GMAC interrupt requires prior programming of the interrupt controller.

The GMAC features 6 interrupt sources. Refer to the table “Peripheral Identifiers” in the section “Peripherals” for the interrupt numbers for GMAC priority queues.

62.6 Functional Description

62.6.1 Media Access Controller

The Media Access Controller (MAC) transmit block takes data from FIFO, adds preamble and, if necessary, pad and frame check sequence (FCS). Both Half Duplex and Full Duplex Ethernet modes of operation are supported. When operating in Half Duplex mode, the MAC transmit block generates data according to the carrier sense multiple access with collision detect (CSMA/CD) protocol. The start of transmission is deferred if carrier sense (CRS) is active. If collision (COL) becomes active during transmission, a jam sequence is asserted and the transmission is retried after a random backoff. The CRS and COL signals have no effect in Full Guplex mode. When operating in Gigabit mode half duplex, both carrier extension and frame bursting are performed in accordance with the IEEE 802.3 standard.

The MAC receive block checks for valid preamble, FCS, alignment and length, and presents received frames to the MAC address checking block and FIFO. Software can configure the GMAC to receive jumbo frames up to 10240 bytes. It can optionally strip CRC from the received frame prior to transfer to FIFO.

The address checker recognizes four specific 48-bit addresses, can recognize four different type ID values, and contains a 64-bit Hash register for matching multicast and unicast addresses as required. It can recognize the broadcast address of all ones and copy all frames. The MAC can also reject all frames that are not VLAN tagged and recognize Wake on LAN events.

The MAC receive block supports offloading of IP, TCP and UDP checksum calculations (both IPv4 and IPv6 packet types supported), and can automatically discard bad checksum frames.

The MAC replaces the timestamp field in PTP 1588 transmit sync frames to support One-Step Clock mode.

The MAC does one-step transparent clock residence time correction for PTP 1588 version 2 transmit sync frames.

62.6.2 1588 Timestamp Unit

The timestamp unit (TSU) consists of a timer and registers to capture the time at which PTP event frames cross the message timestamp point. An interrupt is issued when a capture register is updated.

The 1588 timestamp unit (TSU) is implemented as a 102-bit timer.

The 48 upper bits [101:54] of the timer count seconds and are accessible in the GMAC 1588 Timer Seconds High Register (GMAC_TSH) and GMAC 1588 Timer Seconds Low Register (GMAC_TSL).

The 30 lower bits [53:24] of the timer count nanoseconds and are accessible in the GMAC 1588 Timer Nanoseconds Register (GMAC_TN). The lowest 24 bits [23:0] of the timer count sub-nanoseconds and are accessible in the GMAC 1588 Timer Increment Sub-nanoseconds Register (GMAC_TISUBN).

The 54 lower bits roll over when they have counted to one second. The timer increments by a programmable period (to approximately 58.6 attoseconds resolution) with each clock period and can also be adjusted in 1 ns resolution (incremented or decremented) through APB register accesses.

The clock used can be MCK or GMAC0_TSU that can be connected to a faster clock to increase timestamp accuracy.

The amount by which the timer increments each clock cycle is controlled by the Timer Increment registers (GMAC_TI).

Bits 7:0 are the default increment value in nanoseconds and an additional 24 bits of sub-nanosecond resolution are available using the Timer Increment Subnanoseconds register (GMAC_TISUBN).

If the rest of the register is written with zero, the timer increments by the value in [7:0], plus the value of GMAC_TISUBN, at each clock cycle.

GMAC_TISUBN allows a resolution of approximately 58.6 attoseconds.

Bits 15:8 of GMAC_TI is the alternative increment value in nanoseconds and bits 23:16 are the number of increments after which the alternative increment value is used. If 23:16 are zero, then the alternative increment value will never be used.

Taking the example of 10.2 MHz, there are 102 cycles every ten microseconds or 51 every five microseconds. So a timer with a 10.2 MHz clock source is constructed by incrementing by 98 ns for fifty cycles and then incrementing by 100 ns (98 × 50 + 100 = 5000). This is programmed by setting the 1588 Timer Increment register to 0x00326462.

For a 49.8 MHz clock source it would be 20 ns for 248 cycles followed by an increment of 40 ns (20 × 248 + 40 = 5000) programmed as 0x00F82814.

Having eight bits for the “number of increments” field allows frequencies up to 50 MHz to be supported with 200 kHz resolution.

Without the alternative increment field, the period of the clock would be limited to an integer number of nanoseconds, resulting in supported clock frequencies of 8, 10, 20, 25, 40, 50, 100, 125, 200 and 250 MHz.

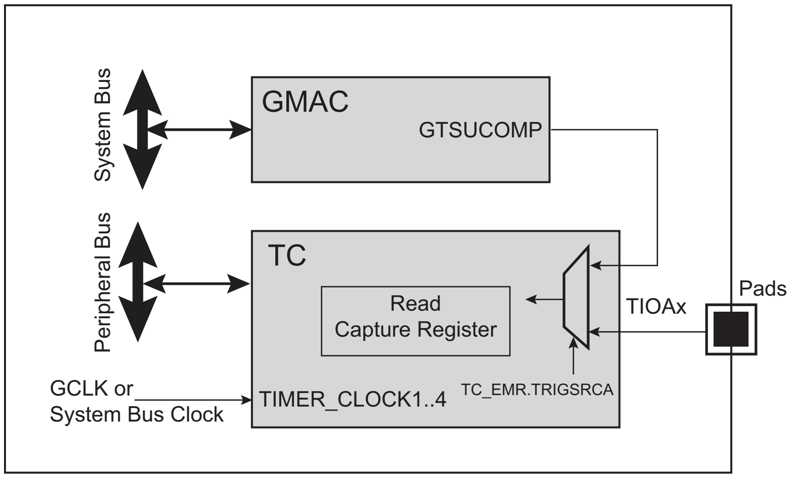
There are additional registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the bits of the seconds value and the upper 22 bits of the nanoseconds value are used.

An interrupt can also be generated (if enabled) when the TSU timer count value and comparison value are equal, mapped to bit 29 of the Interrupt Status register.

A signal (GTSUCOMP) is provided to indicate when the TSU timer count value is equal to the comparison value stored in the TSU timer comparison value registers (0x0DC, 0x0E0, and 0x0E4).

The GTSUCOMP signal is internally routed to one Timer Counter (Refer to the section “Timer Counter (TC)”).

Figure 62-2. GTSUCOMP Internal Connection



62.6.3 Direct Memory Access Interface

The GMAC DMA controller is connected to the MAC FIFO interface and provides a scatter-gather type capability for packet data storage.

The DMA implements packet buffering where dual-port memories are used to buffer multiple frames.

62.6.3.1 Packet Buffer DMA

- Easier to guarantee maximum line rate due to the ability to store multiple frames in the packet buffer, where the number of frames is limited by the amount of packet buffer memory and Ethernet frame size
- Full store and forward
- Partial store and forward programmable options (partial store will cater for shorter latency requirements)
- Support for Transmit TCP/IP checksum offload
- Support for priority queuing
- When a collision on the line occurs during transmission, the packet will be automatically replayed directly from the packet buffer memory rather than having to re-fetch through the system bus (full store and forward ONLY)
- Received error packets are automatically dropped before any of the packet is presented to the system bus (full store and forward ONLY), thus reducing system bus activity
- Supports manual RX packet flush capabilities
- Optional RX packet flush when there is lack of system bus resources

62.6.3.2 Partial Store and Forward Using Packet Buffer DMA

The DMA uses local memories packet buffers, and can be programmed into a low latency mode, known as Partial Store and Forward. This allows for a reduced latency as the full packet is not buffered before forwarding. Note that this option is only available when the device is configured for full duplex operation.

This feature is enabled via the programmable TX and RX Partial Store and Forward registers. When the transmit Partial Store and Forward mode is activated, the transmitter will only begin to forward the packet to the MAC when there is enough packet data stored in the packet buffer. Likewise, when the receive Partial Store and Forward mode is activated, the receiver will only begin to forward the packet to the system bus when enough packet data is stored in the packet buffer. The amount of packet data required to activate the forwarding process is programmable via watermark registers which are located at the same address as the partial store and forward enable bits.

Note that the minimum operational value for the TX partial store and forward watermark is 20. There is no operational limit for the RX partial store and forward watermark. Enabling partial store and forward is a useful means to reduce latency, but there are performance implications.

The GMAC DMA uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in memory. This allows Ethernet packets to be broken up and scattered around the system bus memory space.

62.6.3.3 Receive Buffers

Received frames, optionally including FCS, are written in receive buffers located in system memory. The receive buffer depth is programmable in the range of 64 bytes to 16320 bytes. If received frames are being routed to different priority queues (via the packet inspection screeners – refer to section Priority Queuing in the DMA), it is possible to program different receive buffer depths for each queue. For queue 0, the receive buffer depth is programmed through the DMA Configuration register (offset 0x10). For the other queues, they are programmed in the independent queue configuration registers (starting from offset 0x4a0). The default is 128 bytes.

The start location for each receive buffer is stored in system memory in a list of receive buffer descriptors at an address location pointed to by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured in software using the Receive Buffer Queue Base Address register.

The number of words in each buffer descriptor (BD) is dependent on the operating mode.

Each buffer descriptor (BD) word is defined as 32 bits.

The first two words (Word 0 and Word 1) are used for all BD modes. In Extended Buffer Descriptor modes (GMAC_DCFGR.RXDB_EXTENDED = 1), two BD words are added for 64-bit addressing mode and two BD words are added for timestamp capture.There are therefore either two, four or six BD words in each BD entry depending on the operating mode, and every BD entry has the same number of words.

To summarize:

- Every descriptor is 64 bits wide when the descriptor Timestamp Capture mode is disabled.
- Every descriptor is 128 bits wide when the descriptor Timestamp Capture mode is enabled.

The first is the address of the receive buffer and the second the receive status. If the length of a receive frame exceeds the buffer length, the status word for the used buffer is written with zeroes except for the “start of frame” bit, which is always set for the first buffer in a frame. Bit zero of the address field is written to 1 to show the buffer has been used. The receive buffer manager then reads the location of the next receive buffer and fills that with the next part of the received frame data. Receive buffers are filled until the frame is complete and the final buffer descriptor status word contains the complete frame status. Refer to the table below for details of the receive buffer descriptor list.

Each receive buffer start location is a word address. The start of the first buffer in a frame can be offset by up to three bytes, depending on the value written to bits 14 and 15 of the Network Configuration register. For 64-bit datapaths, the start of the frame can be offset by up to a further four bytes if bit 2 of the buffer start location in the buffer descriptor is set. If the start location of the buffer is offset, the available length of the first buffer is reduced by the corresponding number of bytes.

Table 62-2. Receive Buffer Descriptor Entry

Bit	Function
	Word 0
31:3	Address of beginning of buffer
2	Address [2] of beginning of buffer or in Extended Buffer Descriptor mode (GMAC_DCFGR.RXBD_EXTENDED = 1), indicates a valid timestamp in the BD entry.
1	Wrap—marks last descriptor in receive buffer descriptor list.
0	Ownership—needs to be zero for the GMAC to write data to the receive buffer. The GMAC sets this to one once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
	Word 1
31	Global all ones broadcast address detected
30	Multicast hash match
29	Unicast hash match
28	–

...........continued

Bit	Function
27	Specific Address Register match found, bit 25 and bit 26 indicate which Specific Address Register causes the match.
26:25	Specific Address Register match. Encoded as follows: 00: Specific Address Register 1 match 01: Specific Address Register 2 match 10: Specific Address Register 3 match 11: Specific Address Register 4 match If more than one specific address is matched only one is indicated with priority 4 down to 1.
24	This bit has a different meaning depending on whether RX checksum offloading is enabled. **With RX checksum offloading disabled:** (bit 24 clear in Network Configuration Register) Type ID register match found, bit 22 and bit 23 indicate which type ID register causes the match. **With RX checksum offloading enabled:** (bit 24 set in Network Configuration Register) 0: The frame was not SNAP encoded and/or had a VLAN tag with the Canonical Format Indicator (CFI) bit set. 1: The frame was SNAP encoded and had either no VLAN tag or a VLAN tag with the CFI bit not set.
23:22	This bit has a different meaning depending on whether RX checksum offloading is enabled. **With RX checksum offloading disabled:** (bit 24 clear in Network Configuration) Type ID register match. Encoded as follows: 00: Type ID register 1 match 01: Type ID register 2 match 10: Type ID register 3 match 11: Type ID register 4 match If more than one Type ID is matched only one is indicated with priority 4 down to 1. **With RX checksum offloading enabled:** (bit 24 set in Network Configuration Register) 00: Neither the IP header checksum nor the TCP/UDP checksum was checked. 01: The IP header checksum was checked and was correct. Neither the TCP nor UDP checksum was checked. 10: Both the IP header and TCP checksum were checked and were correct. 11: Both the IP header and UDP checksum were checked and were correct.
21	VLAN tag detected—type ID of 0x8100. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100
20	Priority tag detected—type ID of 0x8100 and null VLAN identifier. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100 and a null VLAN identifier.
19:17	When bit 15 (End of frame) and bit 21 (VLAN tag) are set, these bits represent the VLAN priority. When header/data splitting is enabled (via bit 5 of the DMA configuration register, offset 0x10) bit 17 indicates this descriptor is pointing to the last buffer of the header

...........continued	
Bit	**Function**
16	This bit has a different meaning depending on the state of bit 13 (report bad FCS in bit 16 of word 1 of the receive buffer descriptor) and bit 5 (header/data splitting) of the DMA Configuration register (offset 0x10). When header/data splitting is enabled and this buffer descriptor (BD) is not the last BD of the frame (as indicated in bit 15 of this BD), this bit will indicate that the BD is pointing to a data buffer containing header bytes. When this BD is the last BD of the frame (as indicated in bit 15 of this BD), and bit 13 of the DMA configuration register is set, this bit represents FCS/CRC error. When this BD is the last BD of the frame (as indicated in bit 15 of this BD), and bit 13 of the DMA configuration register is clear, and the received frame is VLAN tagged, this bit represents the Canonical format indicator (CFI).
15	End of frame—when set the buffer contains the end of a frame. If end of frame is not set, then the only valid status bit (unless header/data splitting is enabled) is start of frame (bit 14). If header/data splitting is enabled, then bits 16 and 17 are also valid status bits when this bit is not set.
14	Start of frame—when set the buffer contains the start of a frame. If both bits 15 and 14 are set, the buffer contains a whole frame.
13	This bit has a different meaning depending on whether jumbo frames and ignore FCS modes are enabled. If neither mode is enabled this bit will be zero. **With jumbo frame mode enabled:** (bit 3 set in Network Configuration Register) Additional bit for length of frame (bit[13]), that is concatenated with bits[12:0] **With ignore FCS mode enabled and jumbo frames disabled:** (bit 26 set in Network Configuration Register and bit 3 clear in Network Configuration Register) This indicates per frame FCS status as follows: 0: Frame had good FCS 1: Frame had bad FCS, but was copied to memory as ignore FCS enabled.
12:0	These bits represent the length of the received frame which may or may not include FCS depending on whether FCS discard mode is enabled. **With FCS discard mode disabled:** (bit 17 clear in Network Configuration Register) Least significant 12 bits for length of frame including FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above. **With FCS discard mode enabled:** (bit 17 set in Network Configuration Register) Least significant 12 bits for length of frame excluding FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above.
	Word 2
31:30	Timestamp seconds[1:0] (see Note)
29:0	Timestamp nanoseconds[29:0] (see Note)
	Word 3
31:10	Reserved
9:0	Timestamp seconds[11:2] (see Note)

Note: For details on how to configure Timestamp mode, see GMAC Receive Buffer Data Control Register. The timestamp bits are written back to the last buffer descriptor of a frame only.

To receive frames, the buffer descriptors must be initialized by writing an appropriate address to bits 31:2 in the first word of each list entry. Bit 0 must be written with zero. Bit 1 is the wrap bit and indicates the last entry in the buffer descriptor list.

The start location of the receive buffer descriptor list must be written with the receive buffer queue base address before reception is enabled (receive enable in the Network Control register). Once reception is enabled, any writes to the Receive Buffer Queue Base Address register are ignored. When read, it will return the current pointer position in the descriptor list, though this is only valid and stable when receive is disabled.

If the filter block indicates that a frame should be copied to memory, the receive data DMA operation starts writing data into the receive buffer. If an error occurs, the buffer is recovered.

The receive buffer queue pointer increments by two words after each buffer has been used. It re-initializes to the receive buffer queue base address if any descriptor has its wrap bit set.

As receive buffers are used, the receive buffer manager sets bit zero of the first word of the descriptor to logic one indicating the buffer has been used.

Software should search through the “used” bits in the buffer descriptors to find out how many frames have been received, checking the start of frame and end of frame bits.

When the DMA is configured in the packet buffer Partial Store And Forward mode, received frames are written out to the system bus buffers as soon as enough frame data exists in the packet buffer. For both cases, this may mean several full system bus buffers are used before some error conditions can be detected. If a receive error is detected the receive buffer currently being written will be recovered. Previous buffers will not be recovered. As an example, when receiving frames with cyclic redundancy check (CRC) errors or excessive length, it is possible that a frame fragment might be stored in a sequence of receive buffers. Software can detect this by looking for start of frame bit set in a buffer following a buffer with no end of frame bit set.

To function properly, a 10/100/1000 Ethernet system should have no excessive length frames or frames greater than 128 bytes with CRC errors. Collision fragments will be less than 128 bytes long, therefore it will be a rare occurrence to find a frame fragment in a receive buffer, when using the default value of 128 bytes for the receive buffers size.

When in packet buffer Full Store and Forward mode, only good received frames are written out of the DMA, so no fragments will exist in the system memory buffers due to MAC receiver errors. There is still the possibility of fragments due to DMA errors, for example used bit read on the second buffer of a multi-buffer frame.

If bit zero of the receive buffer descriptor is already set when the receive buffer manager reads the location of the receive buffer, then the buffer has been already used and cannot be used again until software has processed the frame and cleared bit zero. In this case, the “buffer not available” bit in the Receive Status register is set and an interrupt triggered. The Receive Resource Error statistics register is also incremented.

When the DMA is configured in the packet buffer Full Store and Forward mode, the user can optionally select whether received frames should be automatically discarded when no system bus buffer resource is available. This feature is selected via bit 24 of the DMA Configuration register (by default, the received frames are not automatically discarded). If this feature is off, then received packets will remain to be stored in the GMAC local memory packet buffer until the system memory buffer resource next becomes available. This may lead to an eventual packet buffer overflow if packets continue to be received when bit zero (used bit) of the receive buffer descriptor remains set. Note that after a used bit has been read, the receive buffer manager will re-read the location of the receive buffer descriptor every time a new packet is received. When the DMA is not configured in the packet buffer Full Store and Forward mode and a used bit is read, the frame currently being received will be automatically discarded.

When the DMA is configured in the packet buffer Full Store and Forward mode, a receive overrun condition occurs when the receive GMAC local memory packet buffer is full, or because the system bus returns an error. In all other modes, a receive overrun condition occurs when either the system bus was not granted quickly enough, or because of a system bus error, or because a new frame has been detected by the receive block, but the status update or write back for the previous frame has not yet finished. For a receive overrun condition, the receive overrun interrupt is asserted and the buffer currently being written is recovered. The next frame that is received whose address is recognized reuses the buffer.

In any packet buffer mode, a write to bit 18 of GMAC_NCR forces a packet from the external SRAM-based receive packet buffer to be flushed. This feature is only acted upon when the DMA receive channel is not currently writing packet data out to system bus. If the DMA receive channel is active, a write to this bit is ignored.

62.6.3.4 Transmit Buffers

Frames to transmit are stored in one or more transmit buffers located in system memory. Transmit frames can be between 1 and 16384 bytes long, so it is possible to transmit frames longer than the maximum length specified in the IEEE 802.3 standard. It should be noted that zero length buffers are allowed and that the maximum number of buffers permitted for each transmit frame is 128.

The start location for each transmit buffer is stored in memory in a list of transmit buffer descriptors at a location pointed to by the transmit buffer queue pointer. The base address for this queue pointer is set in software using the Transmit Buffer Queue Base Address register.

Each list entry consists of two words.

The number of words in each buffer descriptor (BD) depends on the operating mode.

Each BD word is defined as 32 bits. The first two words (Word 0 and Word 1) are used for all BD modes.

In Extended Buffer Descriptor modes, two BD words are added for timestamp capture. Thus there are either two or four BD words in each BD entry depending on the operating mode, and every BD entry has the same number of words.

To summarize:

- Each descriptor is 64 bits wide when the descriptor timestamp Capture mode is disabled.
- Each descriptor is 128 bits wide when the descriptor timestamp Capture mode is enabled.

The first is the byte address of the transmit buffer and the second containing the transmit control and status. For the packet buffer DMA, the start location for each transmit buffer is a byte address, the bottom bits of the address being used to offset the start of the data from the data-word boundary (i.e., bits 2,1 and 0 are used to offset the address for 64-bit datapaths).

For bus widths of 64 bits, the address of the buffer must be aligned to the correct 64-bit boundary, plus an offset of less than 4 bytes. (Note this alignment restriction in FIFO-based DMA mode only should be sufficient for applications as the main purpose is to allow alignment of the encapsulated IP packet.)

Frames can be transmitted with or without automatic CRC generation. If CRC is automatically generated, pad will also be automatically generated to take frames to a minimum length of 64 bytes. When CRC is not automatically generated (as defined in word 1 of the transmit buffer descriptor), the frame is assumed to be at least 64 bytes long and pad is not generated.

An entry in the transmit buffer descriptor list is described in the table below.

To transmit frames, the buffer descriptors must be initialized by writing an appropriate byte address to bits [31:0] in the first word of each descriptor list entry.

The second word of the transmit buffer descriptor is initialized with control information that indicates the length of the frame, whether or not the MAC is to append CRC and whether the buffer is the last buffer in the frame.

After transmission the status bits are written back to the second word of the first buffer along with the used bit. Bit 31 is the used bit which must be zero when the control word is read if transmission is to take place. It is written to one once the frame has been transmitted. Bits[29:20] indicate various transmit error conditions. Bit 30 is the wrap bit which can be set for any buffer within a frame. If no wrap bit is encountered the queue pointer continues to increment.

The Transmit Buffer Queue Base Address register can only be updated while transmission is disabled or halted; otherwise any attempted write will be ignored. When transmission is halted the transmit buffer queue pointer will maintain its value. Therefore when transmission is restarted the next descriptor read from the queue will be from immediately after the last successfully transmitted frame. while transmit is disabled (bit 3 of the Network Control register set low), the transmit buffer queue pointer resets to point to the address indicated by the Transmit Buffer Queue Base Address register. Note that disabling receive does not have the same effect on the receive buffer queue pointer.

Once the transmit queue is initialized, transmit is activated by writing to the transmit start bit (bit 9) of the Network Control register. Transmit is halted when a buffer descriptor with its used bit set is read, a transmit error occurs, or by writing to the transmit halt bit of the Network Control register. Transmission is suspended if a pause frame is received while the pause enable bit is set in the Network Configuration register. Rewriting the start bit while transmission is active is allowed. This is implemented with TXGO variable which is readable in the Transmit Status register at bit location 3. The TXGO variable is reset when:

- Transmit is disabled.
- A buffer descriptor with its ownership bit set is read.
- Bit 10, THALT, of the Network Control register is written.
- There is a transmit error such as too many retries, late collision (Gigabit mode only) or a transmit underrun.

To set TXGO, write TSTART to the bit 9 of the Network Control register. Transmit halt does not take effect until any ongoing transmit finishes.

If the DMA is configured for packet buffer Partial Store and Forward mode and a collision occurs during transmission of a multi-buffer frame, transmission will automatically restart from the first buffer of the frame. For packet buffer mode, the entire contents of the frame are read into the transmit packet buffer memory, so the retry attempt will be replayed directly from the packet buffer memory rather than having to re-fetch through the system bus.

If a used bit is read midway through transmission of a multi-buffer frame, this is treated as a transmit error. Transmission stops, GTXER is asserted and the FCS will be bad.

If transmission stops due to a transmit error or a used bit being read, transmission restarts from the first buffer descriptor of the frame being transmitted when the transmit start bit is rewritten.

Table 62-3. Transmit Buffer Descriptor Entry

Bit	Function
	Word 0
31:0	Byte address of buffer
	Word 1
31	Used—must be zero for the GMAC to read data to the transmit buffer. The GMAC sets this to one for the first buffer of a frame once it has been successfully transmitted. Software must clear this bit before the buffer can be used again.
30	Wrap—marks last descriptor in transmit buffer descriptor list. This can be set for any buffer within the frame.
29	Retry limit exceeded, transmit error detected
28	Reserved.
27	Transmit frame corruption due to system bus error—set if an error occurs while midway through reading transmit frame from the system bus, including system bus errors and buffers exhausted mid frame (if the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and GTXER asserted). Also set if single frame is too large for configured packet buffer memory size.
26	Late collision, transmit error detected. Late collisions only force this status bit to be set in Gigabit mode.
25:24	Reserved
23	For Extended Buffer Descriptor mode, this bit indicates a timestamp has been captured in the BD. Otherwise Reserved.
22:20	Transmit IP/TCP/UDP checksum generation offload errors: 000: No Error. 001: The Packet was identified as a VLAN type, but the header was not fully complete, or had an error in it. 010: The Packet was identified as a SNAP type, but the header was not fully complete, or had an error in it. 011: The Packet was not of an IP type, or the IP packet was invalidly short, or the IP was not of type IPv4/IPv6. 100: The Packet was not identified as VLAN, SNAP or IP. 101: Non supported packet fragmentation occurred. For IPv4 packets, the IP checksum was generated and inserted. 110: Packet type detected was not TCP or UDP. TCP/UDP checksum was therefore not generated. For IPv4 packets, the IP checksum was generated and inserted. 111: A premature end of packet was detected and the TCP/UDP checksum could not be generated.
19:17	Reserved

...........continued	
Bit	**Function**
16	No CRC to be appended by MAC. When set, this implies that the data in the buffers already contains a valid CRC, hence no CRC or padding is to be appended to the current frame by the MAC. This control bit must be set for the first buffer in a frame and will be ignored for the subsequent buffers of a frame. Note that this bit must be clear when using the transmit IP/TCP/UDP checksum generation offload, otherwise checksum generation and substitution will not occur.
15	Last buffer, when set this bit will indicate the last buffer in the current frame has been reached.
14	Reserved
13:0	Length of buffer
	Word 2
31:30	Timestamp seconds[1:0]
29:0	Timestamp nanoseconds[29:0]
	Word 3
31:10	Reserved
9:0	Timestamp seconds[11:2]

62.6.3.5 DMA Bursting on the System Bus

When performing data transfers, the system bus burst length used can be programmed using bits 4:0 of the DMA Configuration register.

When there is enough space and enough data to be transferred, the programmed fixed length bursts will be used. If there is not enough data or space available, for example when at the beginning or the end of a buffer, single type accesses are used.

The DMA will not terminate a fixed length burst early, unless an error condition occurs on the system bus or if receive or transmit are disabled in the Network Control register.

62.6.3.6 DMA Packet Buffer

The DMA uses packet buffers for both transmit and receive paths. This mode allows multiple packets to be buffered in both transmit and receive directions. This allows the DMA to withstand far greater access latencies on the system bus and make more efficient use of the system bus bandwidth. There are two modes of operation—Full Store and Forward and Partial Store and Forward.

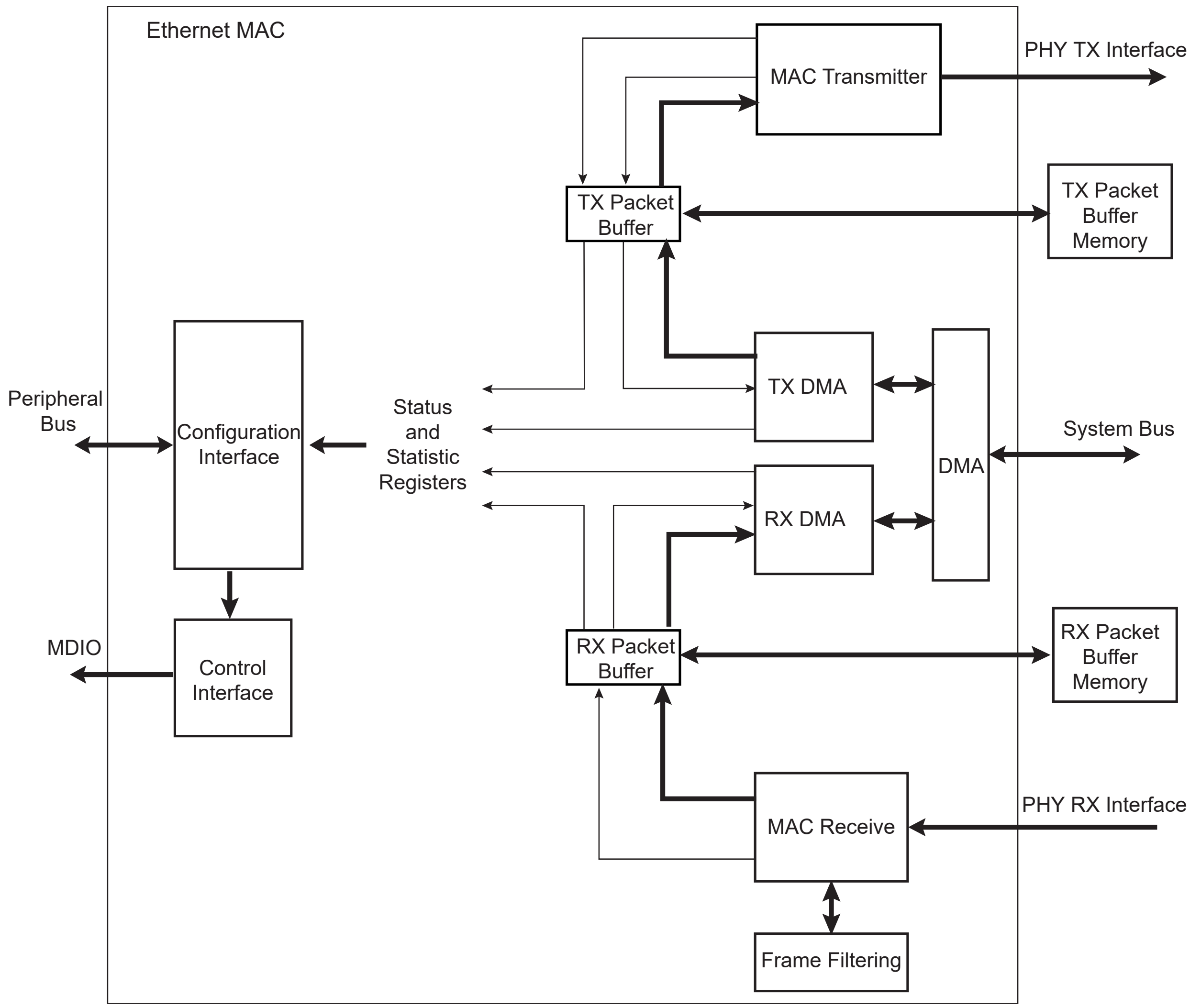
As described in section Partial Store and Forward Using Packet Buffer DMA, the DMA can be programmed into a low latency mode, known as Partial Store and Forward.

When the DMA is in Full Store and Forward mode, Full packet buffering provides the possibility to:

- Discard packets with error on the receive path before they are partially written out of the DMA, thus saving system bus bandwidth and driver processing overhead,,
- Retry collided transmit frames from the buffer, thus saving system bus bandwidth,
- Implement transmit IP/TCP/UDP checksum generation offload.

With the packet buffers included, the structure of the GMAC data paths is shown in the figure below.

Figure 62-3. Data Paths with Packet Buffers Included



62.6.3.7 Transmit Packet Buffer

The transmitter packet buffer will continue attempting to fetch frame data from the system memory until the packet buffer itself is full, at which point it will attempt to maintain its full level.

To accommodate the status and statistics associated with each frame, three words per packet (or two if the GMAC is configured in 64-bit Datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet. Storing the status in the packet buffer memory is required in order to decouple the DMA interface of the buffer from the MAC interface, to update the MAC status/statistics and to generate interrupts in the order in which the packets that they represent were fetched from the system memory.

If any errors occur on the system bus while reading the transmit frame, the fetching of packet data from system memory is halted. The MAC transmitter continues to fetch packet data, thereby emptying the packet buffer and allowing any good non-errored frames to be transmitted successfully. Once these have been fully transmitted, the status/statistics for the errored frame will be updated and software will be informed via an interrupt that a system error occurred. This way, the error is reported in the correct packet order.

The transmit packet buffer will only attempt to read more frame data from the system bus when space is available in the packet buffer memory. If space is not available it must wait until the a packet fetched by the MAC completes transmission and is subsequently removed from the packet buffer memory. Note that if Full Store and Forward mode is active and if a single frame is fetched that is too large for the packet buffer memory, the frame is flushed and the

DMA halted with an error status. This is because a complete frame must be written into the packet buffer before transmission can begin, and therefore the minimum packet buffer memory size should be chosen to satisfy the maximum frame to be transmitted in the application.

In Full Store and Forward mode, once the complete transmit frame is written into the packet buffer memory, a trigger is sent across to the MAC transmitter, which will then begin reading the frame from the packet buffer memory. Since the whole frame is present and stable in the packet buffer memory, an underflow of the transmitter is not possible. The frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in half duplex mode). When this notification is received, the frame is flushed from memory to make room for a new frame to be fetched from the system memory.

In Partial Store and Forward mode, a trigger is sent across to the MAC transmitter as soon as sufficient packet data is available, which will then begin fetching the frame from the packet buffer memory. If, after this point, the MAC transmitter is able to fetch data from the packet buffer faster than the DMA can fill it, an underflow of the transmitter is possible. In this case, the transmission is terminated early, and the packet buffer is completely flushed. Transmission can only be restarted by writing to the transmit START bit.

In Half Duplex mode, the frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in Half Duplex mode). When this notification is received, the frame is flushed from memory to make room for a new frame to be fetched from system memory.

In Full Duplex mode, the frame is removed from the packet buffer on the fly.

Other than underflow, the only MAC related errors that can occur are due to collisions during half duplex transmissions. When a collision occurs the frame still exists in the packet buffer memory so can be retried directly from there. Only once the MAC transmitter has failed to transmit after sixteen attempts is the frame finally flushed from the packet buffer.

62.6.3.8 Receive Packet Buffer

The receive packet buffer stores frames from the MAC receiver along with their status and statistics. Frames with errors are flushed from the packet buffer memory, while good frames are pushed onto the DMA interface.

The receiver packet buffer monitors the FIFO write interface from the MAC receiver and translates the FIFO pushes into packet buffer writes. At the end of the received frame the status and statistics are buffered so that the information can be used when the frame is read out. When programmed in full store and forward mode, if the frame has an error the frame data is immediately flushed from the packet buffer memory allowing subsequent frames to utilise the freed up space. The status and statistics for bad frames are still used to update the GMAC registers.

To accommodate the status and statistics associated with each frame, three words per packet are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet.

The receiver packet buffer will also detect a full condition so that an overflow condition can be detected. If this occurs, subsequent packets are dropped and a receive overflow interrupt is raised.

For Full Store and Forward, the DMA only begins packet fetches once the status and statistics for a frame are available. If the frame has a bad status due to a frame error, the status and statistics are passed on to the GMAC registers. If the frame has a good status, the information is used to read the frame from the packet buffer memory and burst onto the system bus using the DMA buffer management protocol. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

If Partial Store and Forward mode is active, the DMA begins fetching the packet data before the status is available. As soon as the status becomes available, the DMA fetches this information as soon as possible before continuing to fetch the remainder of the frame. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

62.6.3.9 Priority Queueing in the DMA

The DMA by default uses a single transmit and receive queue. This means the list of transmit/receive buffer descriptors point to data buffers associated with a single transmit/receive data stream. The GMAC can select up to 6 priority queues. Each queue has an independent list of buffer descriptors pointing to separate data streams.

The table below gives the memory size associated with each queue:

Table 62-4. Queue Size

Queue Number	Queue Size
5 (highest priority)	4 Kb
4	4 Kb
3	4 Kb
2	4 Kb
1	4 Kb
0 (lowest priority)	4 Kb

In the transmit direction, higher priority queues are always serviced before lower priority queues, with Q0 as lowest priority and Q5 as highest priority. This strict priority scheme requires the user to ensure that high priority traffic is constrained so that lower priority traffic will have required bandwidth. The GMAC DMA will determine the next queue to service by initiating a sequence of buffer descriptor reads interrogating the ownership bits of each. The buffer descriptor corresponding to the highest priority queue is read first. As an example, if the ownership bit of this descriptor is set, then the DMA will progress to reading the 2nd highest priority queue's descriptor. If that ownership bit read of this lower priority queue is set, then the DMA will read the 3rd highest priority queue's descriptor. If all the descriptors return an ownership bit set, then a resource error has occurred, an interrupt is generated and transmission is automatically halted. Transmission can only be restarted by setting the START bit in the Network Control register. The GMAC DMA will need to identify the highest available queue to transmit from when the START bit in the Network Control register is written to and the TX is in a halted state, or when the last word of any packet has been fetched from system memory.

The GMAC transmit DMA maximizes the effectiveness of priority queuing by ensuring that high priority traffic be transmitted as early as possible after being fetched from the system bus. High priority traffic fetched from the system bus is pushed to the MAC layer, depending on traffic shaping being enabled and the associated credit value for that queue, before any lower priority traffic that may pre-exist in the transmit SRAM-based packet buffer. This is achieved by separating the transmit GMAC local memory packet buffer into regions, one region per queue. The size of each region determines the amount of memory space allocated per queue.

For each queue, there is an associated Transmit Buffer Queue Base Address register. For the lowest priority queue (or the only queue when only one queue is selected), the Transmit Buffer Queue Base Address is located at address 0x1C. For all other queues, the Transmit Buffer Queue Base Address registers are located at sequential addresses starting at address 0x440.

In the receive direction each packet is written to system memory data buffers in the order that it is received. For each queue, there is an independent set of receive buffers for each queue. There is therefore a separate Receive Buffer Queue Base Address register for each queue. For the lowest priority queue (or the only queue when only one queue is selected), the Receive Buffer Queue Base Address is located at address 0x18. For all other queues, the Receive Buffer Queue Base Address registers are located at sequential addresses starting at address 0x480. Every received packet will pass through a programmable screening algorithm which will allocate a particular queue to that frame. The user interface to the screeners is through two types of programmable registers:

- Screening Type 1 registers—The module features 4 Screening Type 1 registers (GMAC_ST1RPQ). Screening Type 1 registers hold values to match against specific IP and UDP fields of the received frames. The fields matched against are DS (Differentiated Services field of IPv4 frames), TC (Traffic class field of IPv6 frames) and/or the UDP destination port.
- Screening Type 2 registers—The module features 8 Screening Type 2 registers (GMAC_ST2RPQ). Screening Type 2 registers operate independently of Screening Type 1 registers and offer additional match capabilities. Screening Type 2 allows a screen to be configured that is the combination of all or any of the following comparisons:

1. An enable bit VLAN priority, VLANE. A VLAN priority match will be performed if the VLAN priority enable is set. The extracted priority field in the VLAN header is compared against VLANP in the GMAC_ST2RPQ register itself.

2. An enable bit EtherType, ETHE. The EtherType field I2ETH in GMAC_ST2RPQ maps to one of 4 EtherType match registers, GMAC_ST2ER. The extracted EtherType is compared against GMAC_ST2ER designated by this EtherType field.
3. An enable bit Compare A, COMPAE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, GMAC_ST2CW0R/1R.
4. An enable bit Compare B, COMPBE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, GMAC_ST2CW0R/1R.
5. An enable bit Compare C, COMPCE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, GMAC_ST2CW0R/1R.

Each screener type has an enable bit, a match pattern and a queue number. If a received frame matches on an enabled Screening register, then the frame will be tagged with the queue value in the associated Screening register, and forwarded onto the DMA and subsequently into the external memory associated with that queue. If two screeners are matched, then the one which resides at the lowest register address will take priority so care must be taken on the selection of the screener location.

When the priority queuing feature is enabled, the number of interrupt outputs from the GMAC core is increased to match the number of supported queues. The number of Interrupt Status registers is increased by the same number. Only DMA related events are reported using the individual interrupt outputs, as the GMAC can relate these events to specific queues. All other events generated within the GMAC are reported in the interrupt associated with the lowest priority queue. For the lowest priority queue (or the only queue when only 1 queue is selected), the Interrupt Status register is located at address 0x24. For all other queues, the Interrupt Status register is located at sequential addresses starting at address 0x400.

Note: The address matching is the first level of filtering. If there is a match, the screeners are the next level of filtering for routing the data to the appropriate queue. See MAC Filtering Block for more details.

The additional screening done by the functions Compare A, B, and C each have an enable bit and compare register field. COMPA, COMPB and COMPC in GMAC_ST2RPQ are pointers to a configured offset (OFFSVAL), value (COMPVAL), and mask (MASKVAL). If enabled, the compare is true if the data at the offset into the frame, ANDed with MASKVAL, is equal to the value of COMPVAL ANDed with MASKVAL. A 16-bit word comparison is done. The byte at the offset number of bytes from the index start is compared to bits 7:0 of the configured COMPVAL and MASKVAL. The byte at the offset number of bytes + 1 from the index start is compared to bits 15:8 of the configured COMPVAL and MASKVAL.

The offset value in bytes, OFFSVAL, ranges from 0 to 127 bytes from either the start of the frame, the byte after the EtherType field, the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header. Note the logic to decode the IP header or the TCP/UDP header is reused from the TCP/UDP/IP checksum offload logic and therefore has the same restrictions on use (the main limitation is that IP fragmentation is not supported). Refer to the Checksum Offload for IP, TCP and UDP section of this documentation for further details.

Compare A, B, and C use a common set of 24 GMAC_ST2CW0R/1R registers, thus all COMPA, COMPB and COMPC fields in the registers GMAC_ST2RPQ point to a single pool of 24 GMAC_ST2CW0R/1R registers.

Note that Compare A, B and C together allow matching against an arbitrary 48 bits of data and so can be used to match against a MAC address.

All enabled comparisons are ANDed together to form the overall type 2 screening match.

62.6.4 MAC Transmit Block

The MAC transmitter can operate in either Half Duplex or Full Duplex mode and transmits frames in accordance with the Ethernet IEEE 802.3 standard. In Half Duplex mode, the CSMA/CD protocol of the IEEE 802.3 specification is followed.

A small input buffer receives data through the FIFO interface which, depending on the DMA bus width control bits in the Network Configuration register, will extract data in 32-bit or 64-bit form. All subsequent processing prior to the final output is performed in bytes.

Transmit data can be output using the RGMII/RMII/MII interface.

Frame assembly starts by adding preamble and the start frame delimiter. Data is taken from the transmit FIFO interface a word at a time.

If necessary, padding is added to take the frame length to 60 bytes. CRC is calculated using an order 32-bit polynomial. This is inverted and appended to the end of the frame taking the frame length to a minimum of 64 bytes. If the no CRC bit is set in the second word of the last buffer descriptor of a transmit frame, neither pad nor CRC are appended. The no CRC bit can also be set through the FIFO interface.

In Full Duplex mode (at all data rates), frames are transmitted immediately. Back to back frames are transmitted at least 96 bit times apart to guarantee the interframe gap.

In Half Duplex mode, the transmitter checks carrier sense. If asserted, the transmitter waits for the signal to become inactive, and then starts transmission after the interframe gap of 96 bit times. If the collision signal is asserted during transmission, the transmitter will transmit a jam sequence of 32 bits taken from the data register and then retry transmission after the backoff time has elapsed. If the collision occurs during either the preamble or Start Frame Delimiter (SFD), then these fields will be completed prior to generation of the jam sequence.

The backoff time is based on an XOR of the 10 least significant bits of the data coming from the transmit FIFO interface and a 10-bit pseudo random number generator. The number of bits used depends on the number of collisions seen. After the first collision 1 bit is used, then the second 2 bits and so on up to the maximum of 10 bits. All 10 bits are used above ten collisions. An error will be indicated and no further attempts will be made if 16 consecutive attempts cause collision. This operation is compliant with the description in Clause 4.2.3.2.5 of the IEEE 802.3 standard which refers to the truncated binary exponential backoff algorithm.

In 10/100 mode, both collisions and late collisions are treated identically, and backoff and retry will be performed up to 16 times. When operating in Gigabit mode, late collisions are treated as an exception and transmission is aborted, without retry. This condition is reported in the transmit buffer descriptor word 1 (late collision, bit 26) and also in the Transmit Status register (late collision, bit 7). An interrupt can also be generated (if enabled) when this exception occurs, and bit 5 in the Interrupt Status register will be set.

When operating in Gigabit mode (half duplex) both carrier extension and frame bursting are performed in accordance with the IEEE 802.3 standard. For frames less than 512 bytes carrier extension is used to ensure the minimum slot time is not violated.

Frame bursting is used by the transmitter in Gigabit mode (half duplex) when more than one frame is queued for transmission. The first frame of a burst must be carrier extended (if necessary) to ensure the minimum slot time of 512 bytes is achieved, after which all subsequent frames within the burst must only satisfy the minimum frame length of 64 bytes or greater. Each interframe gap within the burst is filled by the transmitter with carrier extensions, thus ensuring control of the medium is not given up. Several frames may be transmitted up to the burst limit of 65,536 bytes. The transmitter relinquishes control of the medium when there are no more frames queued for transmission or the burst limit is exceeded.

In Gigabit mode any collisions occurring after the minimum slot time for the first frame within a burst are treated as a late collision. The burst is terminated upon this event.

In all modes of operation, if the transmit DMA underruns, a bad CRC is automatically appended using the same mechanism as jam insertion and the GTXER signal is asserted. For a properly configured system this should never occur ; it is also impossible if configured to use the DMA with packet buffers, as the complete frame is buffered in local packet buffer memory.

When bit 28 is set in the Network Configuration register, the Inter Packet Gap (IPG) may be stretched beyond 96 bits depending on the length of the previously transmitted frame and the value written to the IPG Stretch register (GMAC_IPGS). The least significant 8 bits of the IPG Stretch register multiply the previous frame length (including preamble). The next significant 8 bits (+1 so as not to get a divide by zero) divide the frame length to generate the IPG. IPG stretch only works in Full Duplex mode and when bit 28 is set in the Network Configuration register. The IPG Stretch register cannot be used to shrink the IPG below 96 bits.

62.6.5 Transmit Scheduling Algorithm

62.6.5.1 Introduction

The transmit scheduler is responsible for selecting the next queue to be serviced. One of the algorithms can be configured for each queue.

62.6.5.2 802.1Qav Support - Credit-based Shaping

A credit-based shaping algorithm is available on the two highest priority queues and is defined in the standard 802.1Qav: Forwarding and Queuing Enhancements for Time-Sensitive Streams. This allows traffic on these queues to be limited and to allow other queues to transmit.

Traffic shaping is enabled via the CBS (Credit Based Shaping) Control register. This enables a counter which stores the amount of transmit 'credit', measured in bytes that a particular queue has. A queue may only transmit if it has non-negative credit. If a queue has data to send, but is held off from doing as another queue is transmitting, then credit will accumulate in the credit counter at the rate defined in the IdleSlope register (GMAC_CBSISQx) for that queue.

portTransmitRate is the transmission rate, in bits per second, that the underlying MAC service that supports transmission through the Port provides. The value of this parameter is determined by the operation of the MAC.

IdleSlope is the rate of change of increasing credit when waiting to transmit and must be less than the value of the portTransmitRate.

The max value of IdleSlope (or sendSlope) is (portTransmitRate / bits_per_MII_Clock).

In case of 100 Mbps, maximum IdleSlope = (100 Mbps / 4) = 0x17D7840.

When this queue is transmitting, the credit counter is decremented at the rate of sendSlope, which is defined as (portTransmitRate - IdleSlope). A queue can accumulate negative credit when transmitting which will hold off any other transfers from that queue until credit returns to a non-negative value. No transfers are halted when a queue's credit becomes negative; it will accumulate negative credit until the transfer completes.

The highest priority queue always has priority regardless of which queue has the most credit.

62.6.5.3 Fixed Priority

Any of the active queues can be selected as fixed priority and this is the default mode of operation for all queues. The queue index is used as the priority, where a higher index will have a higher priority than a lower index. The scheduler will always attempt to transmit from fixed priority queues with the highest priority (i.e. a fixed priority queue with a high queue index will always take precedence over a priority queue with a lower index).

62.6.5.4 Deficit Weighted Round Robin (DWRR)

Any of the active queues can be selected as DWRR. If DWRR is required, then at least two of the active queues must be selected as DWRR. It must not be used in conjunction with Enhanced Transmission Selection (ETS).

A DWRR enabled queue has lower priority than a fixed priority queue with a higher index.

A DWRR enabled queue has lower priority than a CBS enabled queue.

The DWRR algorithm works by scanning all non-empty queues in sequence. Each queue is allocated a ‘deficit counter’ and an 8-bit weighting (or quantum) value. The value of the deficit counter is the maximum number of bytes that can be sent at the current time.

If the deficit counter of the scanned queue is greater than the length of the packet waiting for transmission, then the packet will be transmitted and the value of the deficit counter is decremented by the packet size. If it is not greater, the scheduler will skip to the next DWRR enabled queue.

If there is insufficient credit to transmit, the queue is simply skipped.

If the queue is empty, the value of the deficit counter is reset to 0.

If all queues have insufficient credit then each tx_clk cycle every queue’s deficit counter is incremented by its quantum value until a queue’s deficit counter obtains sufficient credit to transmit its first queued frame. The higher the quantum value chosen the quicker deficit counter will reach the required value.

If all DWRR queues have the same weighting, then all queues will be granted the same overall bandwidth. The weighting value is stored in four programmable registers starting at offset 0x590.

Note: If fixed priority queues are to be used in conjunction with DWRR, the fixed priority queues must be at a higher index value than the DWRR queues. A consequence of this is that the enabled DWRR queues must form a contiguous set of queues starting from queue 0.

If CBS is also used in conjunction with DWRR, the DWRR queues will share the remaining bandwidth after the CBS allocation has been deducted.

62.6.5.5 Enhanced Transmission Selection (ETS)

The ETS algorithm is defined in IEEE 802.1Qaz: Enhanced Transmission Selection for Bandwidth Sharing between Traffic and allows traffic on specific queues to be bandwidth-limited. Any of the active queues can be selected as ETS. If ETS is required, then at least two of the active queues should be selected as ETS. It must not be used in conjunction with DWRR.

An ETS-enabled queue has lower priority than a CBS-enabled queue or a fixed priority queue with a higher index.

For each ETS-enabled queue, the bandwidth requirement must be configured for each queue as a percentage of total bandwidth (an 8-bit register is used and the sum of values programmed should not exceed decimal 100). This will be the maximum bandwidth to be granted to that queue. The actual scheduling algorithm operates in a round-robin style from lowest indexed queues up to the highest indexed queue in sequence. The bandwidth allocation percentage is stored in programmable registers starting at offset 0x590 – these are the same registers used for DWRR.

If CBS is also used in conjunction with ETS, the sum of the ETS queue percentages should equal the remaining bandwidth after the CBS allocation has been deducted.

Transmit cut-thru must not be enabled if the transmit scheduler is used.

62.6.6 MAC Receive Block

All processing within the MAC receive block is implemented using a 16-bit data path. The MAC receive block checks for valid preamble, FCS, alignment and length, presents received frames to the FIFO interface and stores the frame destination address for use by the address checking block.

If, during the frame reception, the frame is found to be too long, a bad frame indication is sent to the FIFO interface. The receiver logic ceases to send data to memory as soon as this condition occurs.

At end of frame reception the receive block indicates to the DMA block whether the frame is good or bad. The DMA block will recover the current receive buffer if the frame was bad.

Ethernet frames are normally stored in DMA memory complete with the FCS. Setting the FCS remove bit in the network configuration (bit 17) causes frames to be stored without their corresponding FCS. The reported frame length field is reduced by four bytes to reflect this operation.

The receive block signals to the register block to increment the alignment, CRC (FCS), short frame, long frame, jabber or receive symbol errors when any of these exception conditions occur.

If bit 26 is set in the network configuration, CRC errors will be ignored and CRC errored frames will not be discarded, though the Frame Check Sequence Errors statistic register will still be incremented. Additionally, if not enabled for Jumbo Frames mode, then bit 13 of the receiver descriptor word 1 will be updated to indicate the FCS validity for the particular frame. This is useful for applications such as EtherCAT, where individual frames with FCS errors must be identified.

Received frames can be checked for length field error by setting the Length Field Error Frame Discard bit of the Network Configuration register (bit 16). When this bit is set, the receiver compares a frame's measured length with the length field (bytes 13 and 14) extracted from the frame. The frame is discarded if the measured length is shorter. This checking procedure is for received frames between 64 bytes and 1518 bytes in length.

Each discarded frame is counted in the 10-bit Length Field Frame Error statistics register. Frames where the length field is greater than or equal to 0x0600 hex will not be checked.

When operating in Gigabit mode (half duplex), the receiver will discard frames which do not meet the minimal slot time of 512 bytes. If a burst is detected, the first frame is checked to ensure it meets the slot time, but all subsequent frames of the burst are checked to ensure they meet the minimum frame size of 64 bytes.

In Gigabit mode (half duplex), carrier extension errors are detected by the receiver during the minimum slot time, and the frame discarded. An error of this nature causes the Receive Symbol Errors statistic register to be incremented. Carrier extension errors occurring during the inter packet gap period are ignored and have no effect on the statistics.

62.6.7 Checksum Offload for IP, TCP and UDP

The GMAC can be programmed to perform IP, TCP and UDP checksum offloading in both receive and transmit directions, which is enabled by setting bit 24 in the Network Configuration register for receive and bit 11 in the DMA Configuration register for transmit.

IPv4 packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header. TCP and UDP packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header, the data and a conceptual IP pseudo header.

To calculate these checksums in software requires each byte of the packet to be processed. For TCP and UDP this can use a large amount of processing power. Offloading the checksum calculation to hardware can result in significant performance improvements.

For IP, TCP or UDP checksum offload to be useful, the operating system containing the protocol stack must be aware that this offload is available so that it can make use of the fact that the hardware can either generate or verify the checksum.

62.6.7.1 Receiver Checksum Offload

When receive checksum offloading is enabled in the GMAC, the IPv4 header checksum is checked as per RFC 791, where the packet meets the following criteria:

- If present, the VLAN header must be four octets long and the CFI bit must not be set.
- Encapsulation must be RFC 894 Ethernet Type Encoding or RFC 1042 SNAP Encoding.
- IPv4 packet
- IP header is of a valid length

The GMAC also checks the TCP checksum as per RFC 793, or the UDP checksum as per RFC 768, if the following criteria are met:

- IPv4 or IPv6 packet
- Good IP header checksum (if IPv4)
- No IP fragmentation
- TCP or UDP packet

When an IP, TCP or UDP frame is received, the receive buffer descriptor gives an indication if the GMAC was able to verify the checksums. There is also an indication if the frame had SNAP encapsulation. These indication bits will replace the type ID match indication bits when the receive checksum offload is enabled. For details of these indication bits, refer to Receive Buffer Descriptor Entry.

If any of the checksums are verified as incorrect by the GMAC, the packet is discarded and the appropriate statistics counter incremented.

62.6.7.2 Transmitter Checksum Offload

The transmitter checksum offload is only available if the full store and forward mode is enabled. This is because the complete frame to be transmitted must be read into the packet buffer memory before the checksum can be calculated and written back into the headers at the beginning of the frame.

Transmitter checksum offload is enabled by setting bit [11] in the DMA Configuration register. When enabled, it will monitor the frame as it is written into the transmitter packet buffer memory to automatically detect the protocol of the frame. Protocol support is identical to the receiver checksum offload.

For transmit checksum generation and substitution to occur, the protocol of the frame must be recognized and the frame must be provided without the FCS field, by making sure that bit [16] of the transmit descriptor word 1 is clear. If the frame data already had the FCS field, this would be corrupted by the substitution of the new checksum fields.

If these conditions are met, the transmit checksum offload engine will calculate the IP, TCP and UDP checksums as appropriate. Once the full packet is completely written into packet buffer memory, the checksums will be valid and the relevant memory locations will be updated for the new checksum fields as per standard IP/TCP and UDP packet structures.

If the transmitter checksum engine is prevented from generating the relevant checksums, bits [22:20] of the transmitter DMA writeback status will be updated to identify the reason for the error. Note that the frame will still be transmitted but without the checksum substitution, as typically the reason that the substitution did not occur was that the protocol was not recognized.

62.6.8 MAC Filtering Block

The filter block determines which frames should be written to the FIFO interface and on to the DMA.

Whether a frame is passed depends on what is enabled in the Network Configuration register, the contents of the specific address, type and Hash registers and the frame's destination address and type field.

If bit 25 of the Network Configuration register is not set, a frame will not be copied to memory if the GMAC is transmitting in half duplex mode at the time a destination address is received.

Ethernet frames are transmitted a byte at a time, least significant bit first. The first six bytes (48 bits) of an Ethernet frame make up the destination address. The first bit of the destination address, which is the LSB of the first byte of the frame, is the group or individual bit. This is one for multicast addresses and zero for unicast. The all ones address is the broadcast address and a special case of multicast.

The GMAC supports recognition of four specific addresses. Each specific address requires two registers, Specific Address Bottom register and Specific Address Top register. Specific Address Bottom register stores the first four bytes of the destination address and Specific Address Top register contains the last two bytes. The addresses stored can be specific, group, local or universal.

The destination address of received frames is compared against the data stored in the Specific Address registers once they have been activated. The addresses are deactivated at reset or when their corresponding Specific Address Bottom register is written. They are activated when Specific Address Top register is written. If a receive frame address matches an active address, the frame is written to the FIFO interface and on to DMA memory.

Frames may be filtered using the type ID field for matching. Four type ID registers exist in the register address space and each can be enabled for matching by writing a one to the MSB (bit 31) of the respective register. When a frame is received, the matching is implemented as an OR function of the various types of match.

The contents of each type ID register (when enabled) are compared against the length/type ID of the frame being received (e.g., bytes 13 and 14 in non-VLAN and non-SNAP encapsulated frames) and copied to memory if a match is found. The encoded type ID match bits (Word 0, Bit 22 and Bit 23) in the receive buffer descriptor status are set indicating which type ID register generated the match, if the receive checksum offload is disabled.

The reset state of the type ID registers is zero, hence each is initially disabled.

The following example illustrates the use of the address and type ID match registers for a MAC address of 21:43:65:87:A9:CB:

Preamble	55
SFD	D5
DA (Octet 0 - LSB)	21
DA (Octet 1)	43
DA (Octet 2)	65
DA (Octet 3)	87
DA (Octet 4)	A9
DA (Octet 5 - MSB)	CB
SA (LSB)	00[(1)]
SA	00[(1)]
SA	00[(1)]
SA	00[(1)]
SA	00[(1)]
SA (MSB)	00[(1)]
Type ID (MSB)	43
Type ID (LSB)	21

Note:

1. Contains the address of the transmitting device.

The sequence above shows the beginning of an Ethernet frame. Byte order of transmission is from top to bottom as shown. For a successful match to specific address 1, the following address matching registers must be set up:

- Specific Address 1 Bottom register (GMAC_SAB1) (Address 0x088) 0x87654321

- Specific Address 1 Top register (GMAC_SAT1) (Address 0x08C) 0x0000CBA9

For a successful match to the type ID, the following Type ID Match 1 register must be set up:

- Type ID Match 1 register (GMAC_TIDM1) (Address 0x0A8) 0x80004321

62.6.9 Broadcast Address

Frames with the broadcast address of 0xFFFFFFFFFFFF are stored to memory only if the 'no broadcast' bit in the Network Configuration register is set to zero.

62.6.10 Hash Addressing

The hash address register is 64 bits long and takes up two locations in the memory map. The least significant bits are stored in Hash Register Bottom and the most significant bits in Hash Register Top.

The unicast hash enable and the multicast hash enable bits in the Network Configuration register enable the reception of hash matched frames. The destination address is reduced to a 6-bit index into the 64-bit Hash register using the following hash function: The hash function is an XOR of every sixth bit of the destination address.

```
hash_index[05] = da[05] ^ da[11] ^ da[17] ^ da[23] ^ da[29] ^ da[35] ^ da[41] ^ da[47]
hash_index[04] = da[04] ^ da[10] ^ da[16] ^ da[22] ^ da[28] ^ da[34] ^ da[40] ^ da[46]
hash_index[03] = da[03] ^ da[09] ^ da[15] ^ da[21] ^ da[27] ^ da[33] ^ da[39] ^ da[45]
hash_index[02] = da[02] ^ da[08] ^ da[14] ^ da[20] ^ da[26] ^ da[32] ^ da[38] ^ da[44]
hash_index[01] = da[01] ^ da[07] ^ da[13] ^ da[19] ^ da[25] ^ da[31] ^ da[37] ^ da[43]
hash_index[00] = da[00] ^ da[06] ^ da[12] ^ da[18] ^ da[24] ^ da[30] ^ da[36] ^ da[42]
da[0]
```

represents the least significant bit of the first byte received, that is, the multicast/unicast indicator, and `da[47]` represents the most significant bit of the last byte received.

If the hash index points to a bit that is set in the Hash register then the frame will be matched according to whether the frame is multicast or unicast.

A multicast match will be signalled if the multicast hash enable bit is set, `da[0]` is logic 1 and the hash index points to a bit set in the Hash register.

A unicast match will be signalled if the unicast hash enable bit is set, `da[0]` is logic 0 and the hash index points to a bit set in the Hash register.

To receive all multicast frames, the Hash register should be set with all ones and the multicast hash enable bit should be set in the Network Configuration register.

62.6.11 Copy all Frames (Promiscuous Mode)

If the Copy All Frames bit is set in the Network Configuration register then all frames except those that are too long, too short, have FCS errors or have GRXER asserted during reception will be copied to memory. Frames with FCS errors will be copied if bit 26 is set in the Network Configuration register.

62.6.12 Disable Copy of Pause Frames

Pause frames can be prevented from being written to memory by setting the disable copying of pause frames control bit 23 in the Network Configuration register. When set, pause frames are not copied to memory regardless of the Copy All Frames bit, whether a hash match is found, a type ID match is identified or if a destination address match is found.

62.6.13 VLAN Support

The following table describes an Ethernet encoded 802.1Q VLAN tag.

Table 62-5. 802.1Q VLAN Tag

TPID (Tag Protocol Identifier) 16 bits	TCI (Tag Control Information) 16 bits
0x8100	First 3 bits priority, then CFI bit, last 12 bits VID

The VLAN tag is inserted at the 13th byte of the frame adding an extra four bytes to the frame. To support these extra four bytes, the GMAC can accept frame lengths up to 1536 bytes by setting bit 8 in the Network Configuration register.

If the VID (VLAN identifier) is null (0x000) this indicates a priority-tagged frame.

The following bits in the receive buffer descriptor status word give information about VLAN tagged frames:-

- Bit 21 set if receive frame is VLAN tagged (i.e., type ID of 0x8100).
- Bit 20 set if receive frame is priority tagged (i.e., type ID of 0x8100 and null VID). (If bit 20 is set, bit 21 will be set also.)
- Bit 19, 18 and 17 set to priority if bit 21 is set.
- Bit 16 set to CFI if bit 21 is set.

The GMAC can be configured to reject all frames except VLAN tagged frames by setting the discard non-VLAN frames bit in the Network Configuration register.

62.6.14 Wake on LAN Support

The receive block supports Wake on LAN by detecting the following events on incoming receive frames:

- Magic packet
- Address Resolution Protocol (ARP) request to the device IP address
- Specific address 1 filter match
- Multicast hash filter match

These events can be individually enabled through bits [19:16] of the Wake on LAN register. Also, for Wake on LAN detection to occur, receive enable must be set in the Network Control register, however a receive buffer does not have to be available.

In case of an ARP request, specific address 1 or multicast filter events will occur even if the frame is errored. For magic packet events, the frame must be correctly formed and error free.

A magic packet event is detected if all of the following are true:

- Magic packet events are enabled through bit 16 of the Wake on LAN register
- The frame's destination address matches specific address 1
- The frame is correctly formed with no errors
- The frame contains at least 6 bytes of 0xFF for synchronization
- There are 16 repetitions of the contents of Specific Address 1 register immediately following the synchronization

An ARP request event is detected if all of the following are true:

- ARP request events are enabled through bit 17 of the Wake on LAN register
- Broadcasts are allowed by bit 5 in the Network Configuration register
- The frame has a broadcast destination address (bytes 1 to 6)
- The frame has a type ID field of 0x0806 (bytes 13 and 14)
- The frame has an ARP operation field of 0x0001 (bytes 21 and 22)
- The least significant 16 bits of the frame's ARP target protocol address (bytes 41 and 42) match the value programmed in bits[15:0] of the Wake on LAN register

The decoding of the ARP fields adjusts automatically if a VLAN tag is detected within the frame. The reserved value of 0x0000 for the Wake on LAN target address value will not cause an ARP request event, even if matched by the frame.

A specific address 1 filter match event will occur if all of the following are true:

- Specific address 1 events are enabled through bit 18 of the Wake on LAN register
- The frame's destination address matches the value programmed in the Specific Address 1 registers

A multicast filter match event will occur if all of the following are true:

- Multicast hash events are enabled through bit 19 of the Wake on LAN register
- Multicast hash filtering is enabled through bit 6 of the Network Configuration register

- The frame destination address matches against the multicast hash filter
- The frame destination address is not a broadcast

62.6.15 IEEE 1588 Support

IEEE 1588 is a standard for precision time synchronization in local area networks. It works with the exchange of special Precision Time Protocol (PTP) frames. The PTP messages can be transported over IEEE 802.3/Ethernet, over Internet Protocol Version 4 or over Internet Protocol Version 6 as described in the annex of IEEE P1588.D2.1.

The GMAC indicates the message timestamp point (asserted on the start packet delimiter and de-asserted at end of frame) for all frames and the passage of PTP event frames (asserted when a PTP event frame is detected and de-asserted at end of frame).

IEEE 802.1AS is a subset of IEEE 1588. One difference is that IEEE 802.1AS uses the Ethernet multicast address 0180C200000E for sync frame recognition whereas IEEE 1588 does not. GMAC is designed to recognize sync frames with both IEEE 802.1AS and IEEE 1588 addresses and so can support both 1588 and 802.1AS frame recognition simultaneously.

Synchronization between host and client clocks is a two-stage process.

First, the offset between the host and client clocks is corrected by the host sending a sync frame to the client with a follow-up frame containing the exact time the sync frame was sent. Hardware assist modules at the host and client side detect exactly when the sync frame was sent by the host and received by the client. The client then corrects its clock to match the host clock.

Second, the transmission delay between the host and client is corrected. The client sends a delay request frame to the host which sends a delay response frame in reply. Hardware assist modules at the host and client side detect exactly when the delay request frame was sent by the client and received by the host. The client now has enough information to adjust its clock to account for delay. For example, if the client was assuming zero delay, the actual delay will be half the difference between the transmit and receive time of the delay request frame (assuming equal transmit and receive times) because the client clock will be lagging the host clock by the delay time already.

The timestamp is taken when the message timestamp point passes the clock timestamp point. This can generate an interrupt if enabled (GMAC_IER). However, MAC Filtering configuration is needed to actually 'copy' the message to memory. For Ethernet, the message timestamp point is the SFD and the clock timestamp point is the MII interface. (The IEEE 1588 specification refers to sync and delay_req messages as event messages as these require timestamping. These events are captured in the registers GMAC_EFTx and GMAC_EFRx, respectively. Follow up, delay response and management messages do not require timestamping and are referred to as general messages.)

1588 version 2 defines two additional PTP event messages. These are the peer delay request (Pdelay_Req) and peer delay response (Pdelay_Resp) messages. These events are captured in the registers GMAC_PEFTx and GMAC_PEFRx, respectively. These messages are used to calculate the delay on a link. Nodes at both ends of a link send both types of frames (regardless of whether they contain a host or client clock). The Pdelay_Resp message contains the time at which a Pdelay_Req was received and is itself an event message. The time at which a Pdelay_Resp message is received is returned in a Pdelay_Resp_Follow_Up message.

1588 version 2 introduces transparent clocks of which there are two kinds, peer-to-peer (P2P) and end-to-end (E2E). Transparent clocks measure the transit time of event messages through a bridge and amend a correction field within the message to allow for the transit time. P2P transparent clocks additionally correct for the delay in the receive path of the link using the information gathered from the peer delay frames. With P2P transparent clocks delay_req messages are not used to measure link delay. This simplifies the protocol and makes larger systems more stable.

The GMAC recognizes four different encapsulations for PTP event messages:

1. 1588 version 1 (UDP/IPv4 multicast)
2. 1588 version 2 (UDP/IPv4 multicast)
3. 1588 version 2 (UDP/IPv6 multicast)
4. 1588 version 2 (Ethernet multicast)

Table 62-6. Example of Sync Frame in 1588 Version 1 Format

Frame Segment	Value
Preamble/SFD	55555555555555D5

...........continued

Frame Segment	Value
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	–
UDP (Octet 23)	11
IP stuff (Octets 24–29)	–
IP DA (Octets 30–32)	E00001
IP DA (Octet 33)	81 or 82 or 83 or 84
Source IP port (Octets 34–35)	–
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–42)	–
Version PTP (Octet 43)	01
Other stuff (Octets 44–73)	–
Control (Octet 74)	00
Other stuff (Octets 75–168)	–

Table 62-7. Example of Delay Request Frame in 1588 Version 1 Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	–
UDP (Octet 23)	11
IP stuff (Octets 24–29)	–
IP DA (Octets 30–32)	E00001
IP DA (Octet 33)	81 or 82 or 83 or 84
Source IP port (Octets 34–35)	–
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–42)	–
Version PTP (Octet 43)	01
Other stuff (Octets 44–73)	–
Control (Octet 74)	01
Other stuff (Octets 75–168)	–

For 1588 version 1 messages, sync and delay request frames are indicated by the GMAC if the frame type field indicates TCP/IP, UDP protocol is indicated, the destination IP address is 224.0.1.129/130/131 or 132, the destination UDP port is 319 and the control field is correct.

The control field is 0x00 for sync frames and 0x01 for delay request frames.

For 1588 version 2 messages, the type of frame is determined by looking at the message type field in the first byte of the PTP frame. Whether a frame is version 1 or version 2 can be determined by looking at the version PTP field in the second byte of both version 1 and version 2 PTP frames.

In version 2 messages sync frames have a message type value of 0x0, delay_req have 0x1, Pdelay_Req have 0x2 and Pdelay_Resp have 0x3.

Table 62-8. Example of Sync Frame in 1588 Version 2 (UDP/IPv4) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	–
UDP (Octet 23)	11
IP stuff (Octets 24–29)	–
IP DA (Octets 30–33)	E0000181
Source IP port (Octets 34–35)	–
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–41)	–
Message type (Octet 42)	00
Version PTP (Octet 43)	02

Table 62-9. Example of Pdelay_Req Frame in 1588 Version 2 (UDP/IPv4) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	–
UDP (Octet 23)	11
IP stuff (Octets 24–29)	–
IP DA (Octets 30–33)	E000006B
Source IP port (Octets 34–35)	–
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–41)	–
Message type (Octet 42)	02

...........continued

Frame Segment	Value
Version PTP (Octet 43)	02

Table 62-10. Example of Sync Frame in 1588 Version 2 (UDP/IPv6) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	86dd
IP stuff (Octets 14–19)	–
UDP (Octet 20)	11
IP stuff (Octets 21–37)	–
IP DA (Octets 38–53)	FF0X00000000018
Source IP port (Octets 54–55)	–
Dest IP port (Octets 56–57)	013F
Other stuff (Octets 58–61)	–
Message type (Octet 62)	00
Other stuff (Octets 63–93)	–
Version PTP (Octet 94)	02

Table 62-11. Example of Pdelay_Resp Frame in 1588 Version 2 (UDP/IPv6) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	–
SA (Octets 6–11)	–
Type (Octets 12–13)	86dd
IP stuff (Octets 14–19)	–
UDP (Octet 20)	11
IP stuff (Octets 21–37)	–
IP DA (Octets 38–53)	FF0200000000006B
Source IP port (Octets 54–55)	–
Dest IP port (Octets 56–57)	013F
Other stuff (Octets 58–61)	–
Message type (Octet 62)	03
Other stuff (Octets 63–93)	–
Version PTP (Octet 94)	02

For the multicast address 011B19000000 sync and delay request frames are recognized depending on the message type field, 00 for sync and 01 for delay request.

Table 62-12. Example of Sync Frame in 1588 Version 2 (Ethernet Multicast) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	011B19000000
SA (Octets 6–11)	–
Type (Octets 12–13)	88F7
Message type (Octet 14)	00
Version PTP (Octet 15)	02

Pdelay request frames need a special multicast address so they can pass through ports blocked by the spanning tree protocol. For the multicast address 0180C200000E sync, Pdelay_Req and Pdelay_Resp frames are recognized depending on the message type field, 00 for sync, 02 for pdelay request and 03 for pdelay response.

Table 62-13. Example of Pdelay_Req Frame in 1588 Version 2 (Ethernet Multicast) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	0180C200000E
SA (Octets 6–11)	–
Type (Octets 12–13)	88F7
Message type (Octet 14)	00
Version PTP (Octet 15)	02

62.6.16 MAC 802.3 Pause Frame Support

Note: See Clause 31, and Annex 31A and 31B of the IEEE standard 802.3 for a full description of MAC 802.3 pause operation.

The following table shows the start of a MAC 802.3 pause frame.

Table 62-14. Start of an 802.3 Pause Frame

Address		Type (MAC Control Frame)	Pause	
Destination	Source		Opcode	Time
0x0180C2000001	6 bytes	0x8808	0x0001	2 bytes

The GMAC supports both hardware controlled pause of the transmitter, upon reception of a pause frame, and hardware generated pause frame transmission.

62.6.16.1 802.3 Pause Frame Reception

Bit 13 of the Network Configuration register is the pause enable control for reception. If this bit is set, transmission pauses if a non zero pause quantum frame is received.

If a valid pause frame is received, then the Pause Time register is updated with the new frame's pause time, regardless of whether a previous pause frame is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register.

Once the Pause Time register is loaded and the frame currently being transmitted has been sent, no new frames are transmitted until the pause time reaches zero. The loading of a new pause time, and hence the pausing of transmission, only occurs when the GMAC is configured for full duplex operation. If the GMAC is configured for half duplex there will be no transmission pause, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address 1 register or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0001.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. 802.3 Pause frames that are received after Priority-based Flow Control (PFC) has been negotiated will also be discarded. Valid pause frames received will increment the Pause Frames Received statistic register.

The Pause Time register decrements every 512 bit times once transmission has stopped. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GTXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

62.6.16.2 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 or bit 12 of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address 1 register
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A Pause Quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 11 is written with a one, the pause quantum will be taken from the Transmit Pause Quantum register. The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.
- If bit 12 is written with a one, the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and only the statistics register Pause Frames Transmitted is incremented.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

62.6.17 MAC PFC Priority-based Pause Frame Support

Note: Refer to the 802.1Qbb standard for a full description of priority-based pause operation.

The following table shows the start of a Priority-based Flow Control (PFC) pause frame.

Table 62-15. Start of a PFC Pause Frame

Address		Type (Mac Control Frame)	Pause Opcode	Priority Enable Vector	Pause Time
Destination	Source				
0x0180C2000001	6 bytes	0x8808	0x1001	2 bytes	8 × 2 bytes

The GMAC supports PFC priority-based pause transmission and reception. Before PFC pause frames can be received, bit 16 of the Network Control register must be set.

62.6.17.1 PFC Pause Frame Reception

The ability to receive and decode priority-based pause frames is enabled by setting bit 16 of the Network Control register. When this bit is set, the GMAC will match either classic 802.3 pause frames or PFC priority-based pause frames. Once a priority-based pause frame has been received and matched, then from that moment on the GMAC will only match on priority-based pause frames (this is an 802.1Qbb requirement, known as PFC negotiation). Once priority-based pause has been negotiated, any received 802.3x format pause frames will not be acted upon.

If a valid priority-based pause frame is received then the GMAC will decode the frame and determine which, if any, of the eight priorities require to be paused. Up to eight Pause Time registers are then updated with the eight pause times extracted from the frame regardless of whether a previous pause operation is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register. The loading of a new pause time only occurs when the GMAC is configured for full duplex operation. If the GMAC is configured for half duplex, the pause time counters will not be loaded, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address 1 register or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0101.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. Valid pause frames received will increment the Pause Frames Received Statistic register.

The Pause Time registers decrement every 512 bit times immediately following the PFC frame reception. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GRXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

62.6.17.2 PFC Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit priority-based pause frame bit of the Network Control register. If bit 17 of the Network Control register is written with logic 1, a PFC pause frame will be transmitted providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register. When bit 17 of the Network Control register is set, the fields of the priority-based pause frame will be built using the values stored in the Transmit PFC Pause register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address 1 register
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 01-01
- A priority enable vector taken from Transmit PFC Pause register
- 8 Pause Quantum registers
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The Pause Quantum registers used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 17 of the Network Control register is written with a one, then the priority enable vector of the priority-based pause frame will be set equal to the value stored in the Transmit PFC Pause register [7:0]. For each entry equal to zero in the Transmit PFC Pause register [15:8], the pause quantum field of the pause frame associated with that entry will be taken from the Transmit Pause Quantum register. For each entry equal to one in the Transmit PFC Pause register [15:8], the pause quantum associated with that entry will be zero.

- The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

PFC Pause frames can also be transmitted by the MAC using normal frame transmission methods.

62.6.18 Energy-efficient Ethernet Support

IEEE 802.3az adds support for energy efficiency to Ethernet. These are the key features of 802.3az:

- Allows a system's transmit path to enter a Low-Power mode if there is nothing to transmit.
- Allows a PHY to detect whether its link partner's transmit path is in Low-Power mode, therefore allowing the system's receive path to enter Low-Power mode.
- Link remains up during lower power mode and no frames are dropped.
- Asymmetric, one direction can be in Low-Power mode while the other is transmitting normally.
- LPI (Low Power Idle) signaling is used to control entry and exit to and from Low-Power modes.
- LPI signaling can only take place if both sides have indicated support for it through auto-negotiation.

These are the key features of 802.3az operation:

- Low-power control is done at the MII (reconciliation sublayer).
- As an architectural convenience in writing the 802.3az it is assumed that transmission is deferred by asserting carrier sense, in practice it will not be done this way. This system will know when it has nothing to transmit and only enter Low-Power mode when it is not transmitting.
- LPI should not be requested unless the link has been up for at least one second.
- LPI is signaled on the transmit path by asserting 0x01 on txd with tx_en low and tx_er high.
- A PHY on seeing LPI requested on the MII will send the sleep signal before going quiet. After going quiet it will periodically transmit refresh signals.
- The sleep, quiet and refresh periods are defined in Table 78-2 of 802.3az. For 1000BASE-X the sleep period is 20 microseconds, the quiet period 2.5 milliseconds and the refresh period 20 microseconds.
- 1000BASE-X is required to go quiet after sleep is signaled. The easiest way to do this is to write to a control register to disable transmit in the SerDes.
- LPI mode ends by transmitting normal idle for the wake time. There is a default time for this but it can be adjusted in software using the Link Layer Discovery Protocol (LLDP) described in Clause 79 of 802.3az.
- LPI is indicated at the receive side when sleep and refresh signaling has been detected.

62.6.19 LPI Operation in the GMAC

It is best to use firmware to control LPI. LPI operation happens at the system level. Firmware gives maximum control and flexibility of operation. LPI operation is straightforward and firmware should be capable of responding within the required timeframes.

Auto-negotiation:

1. Indicate EEE capability using next page auto-negotiation.

For the transmit path:

1. If the link has been up for 1 second and there is nothing being transmitted, write to the TXLPIEN bit in the Network Control register.
2. If connected to 1000BASE-T PHY using RGMII, there is nothing more to do.
3. If connected to a backplane using a 1000BASE-KX PHY, use firmware to periodically disable the SerDes transmit path. (Write to bit 1.160.0 for 1000BASE-KX.)
4. Wake up by clearing the TXLPIEN bit in the Network Control register.

For the receive path:

1. Enable RXLPISBC bit in GMAC_IER. The bit RXLPIS is set in Network Status Register triggering an interrupt.
2. Wait for an interrupt to indicate that LPI has been received.
3. Disable relevant parts of the receive path if desired but keep the PCS and SerDes active.

4. The RXLPIS bit in Network Status Register gets cleared to indicate that regular idle has been received. This triggers an interrupt.
5. Re-enable the receive path.

62.6.20 PHY Interface

Different PHY interfaces are supported by the Gigabit Ethernet MAC:

- MII
- RMII
- RGMIIv1.3

The RGMII should only be used for 1000 Mbps operation. The MII interface is provided for 10/100 operation and uses txd[3:0] and rxd[3:0]. The RMII interface is provided for 10/100 operation and uses txd[1:0] and rxd[1:0].

62.6.21 10/100/1000 Operation

The Gigabit Mode Enable (GBE) bit in the Network Configuration register selects between 10/100 Mbps Ethernet operation and 1000 Mbps operation. The Speed (SPD) bit in the Network Configuration register is used to select between 10 Mbps and 100 Mbps operation.

62.6.22 Jumbo Frames

The Jumbo Frame Size (JFRAME) bit in the Network Configuration register allows the GMAC, in its default configuration, to receive jumbo frames up to 10240 bytes in size. This operation does not form part of the IEEE 802.3 specification and is normally disabled. When jumbo frames are enabled, frames received with a frame size greater than 10240 bytes are discarded.

62.7 Programming Interface

62.7.1 Initialization

62.7.1.1 Configuration

Initialization of the GMAC configuration (e.g., loop back mode, frequency ratios) must be done while the transmit and receive circuits are disabled. See the description of the Network Control register and Network Configuration register earlier in this document.

To change loop back mode, the following sequence of operations must be followed:

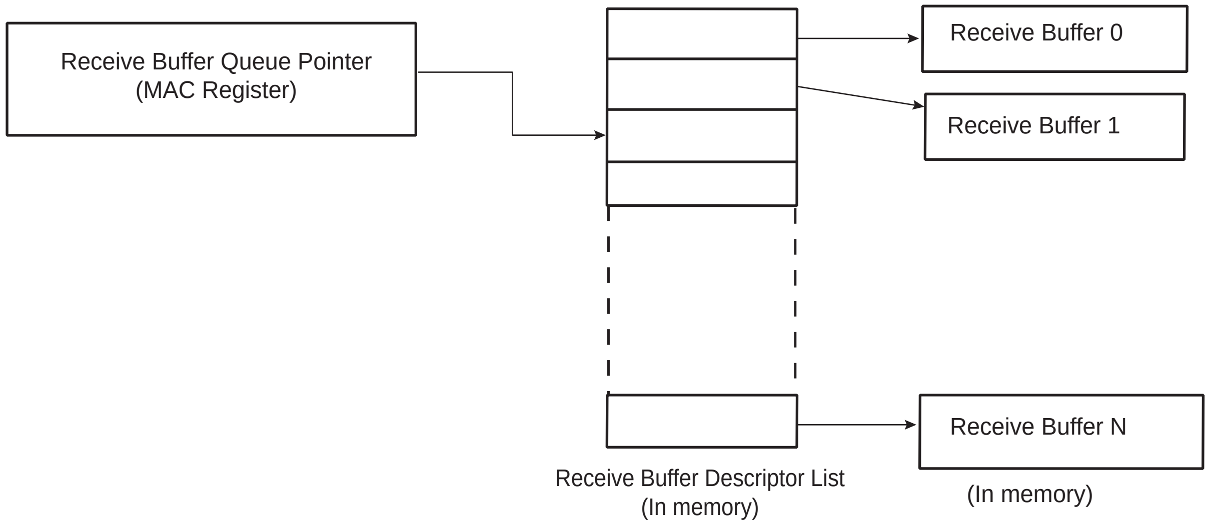
1. Write to Network Control register to disable transmit and receive circuits.
2. Write to Network Control register to change loop back mode.
3. Write to Network Control register to re-enable transmit or receive circuits.
   Note: These writes to the Network Control register cannot be combined in any way.

62.7.1.2 Receive Buffer List

Receive data is written to areas of data (i.e., buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (receive buffer queue) is a sequence of descriptor entries as defined in Receive Buffer Descriptor Entry.

The Receive Buffer Queue Pointer register points to this data structure.

Figure 62-4. Receive Buffer List



To create the list of buffers:

1. Allocate a number (N) of buffers of X bytes in system memory, where X is the DMA buffer length programmed in the DMA Configuration register.
2. Allocate an area 8N bytes for the receive buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by GMAC, i.e., bit 0 of word 0 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 1 in word 0 set to 1).
4. Write address of receive buffer descriptor list and control information to GMAC register receive buffer queue pointer
5. The receive circuits can then be enabled by writing to the address recognition registers and the Network Control register.
   Note: The queue pointers must be initialized and point to USED descriptors for all queues including those not intended for use.

62.7.1.3 Transmit Buffer List

Transmit data is read from areas of data (the buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (Transmit Buffer Queue) is a sequence of descriptor entries as defined in Transmit Buffer Descriptor Entry.

The Transmit Buffer Queue Pointer register points to this data structure.

To create this list of buffers:

1. Allocate a number (N) of buffers of between 1 and 2047 bytes of data to be transmitted in system memory. Up to 128 buffers per frame are allowed.
2. Allocate an area 8N bytes for the transmit buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by GMAC, i.e., bit 31 of word 1 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 30 in word 1 set to 1).
4. Write address of transmit buffer descriptor list and control information to GMAC register transmit buffer queue pointer.
5. The transmit circuits can then be enabled by writing to the Network Control register.
   Note: The queue pointers must be initialized and point to USED descriptors for all queues including those not intended for use.

62.7.1.4 Address Matching

The GMAC Hash register pair and the four Specific Address register pairs must be written with the required values. Each register pair comprises of a bottom register and top register, with the bottom register being written first. The address matching is disabled for a particular register pair after the bottom register has been written and re-enabled when the top register is written. Each register pair may be written at any time, regardless of whether the receive circuits are enabled or disabled.

As an example, to set Specific Address 1 register to recognize destination address 21:43:65:87:A9:CB, the following values are written to Specific Address 1 Bottom register and Specific Address 1 Top register:

- Specific Address 1 Bottom register bits 31:0 (0x98): 0x8765_4321.
- Specific Address 1 Top register bits 31:0 (0x9C): 0x0000_CBA9.

Note: The address matching is the first level of filtering. If there is a match, the screeners are the next level of filtering for routing the data to the appropriate queue. See Priority Queueing in the DMA for more details.

62.7.1.5 PHY Maintenance

The PHY Maintenance register is implemented as a shift register. Writing to the register starts a shift operation which is signaled as complete when bit 2 (IDLE) is set in the Network Status register (about 2000 MCK cycles later when bits 18:16 are set to ‘2’ in the Network Configuration register). An interrupt is generated as this bit is set.

During this time, the MSB of the register is output on the GMDIO pin and the LSB updated from the GMDIO pin with each Management Data Clock (MDC) cycle. This causes the transmission of a PHY management frame on GMDIO pin. Refer to section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation will return the current contents of the shift register. At the end of the management operation the bits will have shifted back to their original locations. For a read operation the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The Management Data Clock (MDC) should not toggle faster than 2.5 MHz (minimum period of 400 ns), as defined by the IEEE 802.3 standard. MDC is generated by dividing down MCK. Three bits in the Network Configuration register determine by how much MCK should be divided to produce MDC.

62.7.1.6 Interrupts

There are multiple interrupt sources that are detected to drive multiple interrupt lines. Depending on the overall system design this may be passed through a further level of interrupt collection (interrupt controller). On receipt of the interrupt signal, the CPU enters the interrupt handler. Refer to the device interrupt controller documentation to identify that it is the GMAC that is generating the interrupt. To ascertain which interrupt, read the Interrupt Status register. Note that in the default configuration this register will clear itself after being read, though this may be configured to be write-one-to-clear if desired.

At reset all interrupts are disabled. To enable an interrupt, write to Interrupt Enable register with the pertinent interrupt bit set to 1. To disable an interrupt, write to Interrupt Disable register with the pertinent interrupt bit set to 1. To check whether an interrupt is enabled or disabled, read Interrupt Mask register. If the bit is set to 1, the interrupt is disabled.

62.7.1.7 Transmitting Frames

The procedure to set up a frame for transmission is the following:

1. Enable transmit in the Network Control register.
2. Allocate an area of system memory for transmit data. This does not have to be contiguous, varying byte lengths can be used if they conclude on byte borders.
3. Set-up the transmit buffer list by writing buffer addresses to word zero of the transmit buffer descriptor entries and control and length to word one.
4. Write data for transmission into the buffers pointed to by the descriptors.
5. Write the address of the first buffer descriptor to transmit buffer descriptor queue pointer.
6. Enable appropriate interrupts.
7. Write to the transmit start bit (TSTART) in the Network Control register.

62.7.1.8 Receiving Frames

When a frame is received and the receive circuits are enabled, the GMAC checks the address and, in the following cases, the frame is written to system memory:

- If it matches one of the four Specific Address registers.
- If it matches one of the four Type ID registers.
- If it matches the hash address function.
- If it is a broadcast address (0xFFFFFFFFFFFF) and broadcasts are allowed.
- If the GMAC is configured to “copy all frames”.

The register receive buffer queue pointer points to the next entry in the receive buffer descriptor list and the GMAC uses this as the address in system memory to write the frame to.

Once the frame has been completely and successfully received and written to system memory, the GMAC then updates the receive buffer descriptor entry (see Receive Buffer Descriptor Entry) with the reason for the address match and marks the area as being owned by software. Once this is complete, a receive complete interrupt is set. Software is then responsible for copying the data to the application area and releasing the buffer (by writing the ownership bit back to 0).

If the GMAC is unable to write the data at a rate to match the incoming frame, then a receive overrun interrupt is set. If there is no receive buffer available, i.e., the next buffer is still owned by software, a receive buffer not available interrupt is set. If the frame is not successfully received, a statistics register is incremented and the frame is discarded without informing software.

62.7.2 Statistics Registers

Statistics registers are described beginning with GMAC Octets Transmitted Low Register and ending with GMAC UDP Checksum Errors Register.

The statistics register block begins at 0x100 and runs to 0x1B0, and comprises the registers listed below.

Octets Transmitted Low Register	Broadcast Frames Received Register
Octets Transmitted High Register	Multicast Frames Received Register
Frames Transmitted Register	Pause Frames Received Register
Broadcast Frames Transmitted Register	64 Byte Frames Received Register
Multicast Frames Transmitted Register	65 to 127 Byte Frames Received Register
Pause Frames Transmitted Register	128 to 255 Byte Frames Received Register
64 Byte Frames Transmitted Register	256 to 511 Byte Frames Received Register
65 to 127 Byte Frames Transmitted Register	512 to 1023 Byte Frames Received Register
128 to 255 Byte Frames Transmitted Register	1024 to 1518 Byte Frames Received Register
256 to 511 Byte Frames Transmitted Register	1519 to Maximum Byte Frames Received Register
512 to 1023 Byte Frames Transmitted Register	Undersize Frames Received Register
1024 to 1518 Byte Frames Transmitted Register	Oversize Frames Received Register
Greater Than 1518 Byte Frames Transmitted Register	Jabbers Received Register
Transmit Underruns Register	Frame Check Sequence Errors Register
Single Collision Frames Register	Length Field Frame Errors Register
Multiple Collision Frames Register	Receive Symbol Errors Register
Excessive Collisions Register	Alignment Errors Register
Late Collisions Register	Receive Resource Errors Register
Deferred Transmission Frames Register	Receive Overrun Register
Carrier Sense Errors Register	IP Header Checksum Errors Register
Octets Received Low Register	TCP Checksum Errors Register
Octets Received High Register	UDP Checksum Errors Register
Frames Received Register	

These registers reset to zero on a read and stick at all ones when they count to their maximum value. They should be read frequently enough to prevent loss of data.

The receive statistics registers are only incremented when the receive enable bit (RXEN) is set in the Network Control register.

Once a statistics register has been read, it is automatically cleared. When reading the Octets Transmitted and Octets Received registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

62.8 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	GMAC_NCR	31:24		IFGQAVCRED		MIIONRGMII	OSSCORR	EXTSELRQEN	PFCCTL	OSSMODE
		23:16		STUDPOFFSET		PTPUNIENA	TXLPIEN	FNP	TXPBPF	ENPBPR
		15:8	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
		7:0	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
0x04	GMAC_NCFGR	31:24		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
		23:16	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
		15:8	RXBUFO[1:0]		PEN	RTY		GBE		MAXFS
		7:0	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
0x08	GMAC_NSR	31:24								
		23:16								
		15:8								SB_PEND_XFER
		7:0	RXLPIS	PFCPAUSN				IDLE	MDIO	
0x0C	GMAC_UR	31:24								
		23:16								
		15:8								
		7:0		HDFLCTLEN				REFCLK	MIM[1:0]	
0x10	GMAC_DCFGR	31:24			TXBD_EXTENDED	RXBD_EXTENDED				DDRP
		23:16	DRBS[7:0]							
		15:8				INFLASTEN	TXCOEN	TXPBMS	RXBMS[1:0]	
		7:0	ESPA	ESMA		FBLDO[4:0]				
0x14	GMAC_TSR	31:24								
		23:16								
		15:8								HRESP
		7:0	LCO		TXCOMP	TFC	TXGO	RLE	COL	UBR
0x18	GMAC_RBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							RXQDIS
0x1C	GMAC_TBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							TXQDIS
0x20	GMAC_RSR	31:24								
		23:16								
		15:8								
		7:0					HNO	RXOVR	REC	BNA
0x24	GMAC_ISR	31:24			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8		PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x28	GMAC_IER	31:24			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	GMAC_IDR	31:24			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x30	GMAC_IMR	31:24			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x34	GMAC_MAN	31:24	WZO	CLTTO	OP[1:0]		PHYA[4:1]			
		23:16	PHYA[0]	REGA[4:0]					WTN[1:0]	
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x38	GMAC_RPQ	31:24								
		23:16								
		15:8	RPQ[15:8]							
		7:0	RPQ[7:0]							
0x3C	GMAC_TPQ	31:24	P1TPQ[15:8]							
		23:16	P1TPQ[7:0]							
		15:8	TPQ[15:8]							
		7:0	TPQ[7:0]							
0x40	GMAC_TPSF	31:24	ENTXP							
		23:16								
		15:8						TPB1ADR[10:8]		
		7:0	TPB1ADR[7:0]							
0x44	GMAC_RPSF	31:24	ENRXP							
		23:16								
		15:8						RPB1ADR[10:8]		
		7:0	RPB1ADR[7:0]							
0x48	GMAC_RJFML	31:24								
		23:16								
		15:8			FML[13:8]					
		7:0	FML[7:0]							
0x4C ... 0x53	Reserved									
0x54	GMAC_AMP	31:24								
		23:16								USE_FROM
		15:8	AW2W_MAX_PIPELINE[7:0]							
		7:0	AR2R_MAX_PIPELINE[7:0]							
0x58 ... 0x5B	Reserved									
0x5C	GMAC_INTM	31:24								
		23:16	TXINTMOD[7:0]							
		15:8								
		7:0	RXINTMOD[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x60	GMAC_SYSWT	31:24								
		23:16								
		15:8	SYSWKUPTIME[15:8]							
		7:0	SYSWKUPTIME[7:0]							
0x64 ... 0x7F	Reserved									
0x80	GMAC_HRB	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x84	GMAC_HRT	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x88	GMAC_SAB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x8C	GMAC_SAT1	31:24								
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x90	GMAC_SAB2	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x94	GMAC_SAT2	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x98	GMAC_SAB3	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x9C	GMAC_SAT3	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xA0	GMAC_SAB4	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

..........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0xA4	GMAC_SAT4	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xA8	GMAC_TIDM1	31:24	ENID1							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0xAC	GMAC_TIDM2	31:24	ENID2							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0xB0	GMAC_TIDM3	31:24	ENID3							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0xB4	GMAC_TIDM4	31:24	ENID4							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0xB8	GMAC_WOL	31:24								
		23:16					MTI	SA1	ARP	MAG
		15:8	IP[15:8]							
		7:0	IP[7:0]							
0xBC	GMAC_IPGS	31:24								
		23:16								
		15:8	FL[15:8]							
		7:0	FL[7:0]							
0xC0	GMAC_SVLAN	31:24	ESVLAN							
		23:16								
		15:8	VLAN_TYPE[15:8]							
		7:0	VLAN_TYPE[7:0]							
0xC4	GMAC_TPFCP	31:24								
		23:16								
		15:8	PQ[7:0]							
		7:0	PEV[7:0]							
0xC8	GMAC_SAMB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xCC	GMAC_SAMT1	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xD0	GMAC_AMRX	31:24	MSBADDR[3:0]							
		23:16								
		15:8								
		7:0					MSBADDRMSK[3:0]			
0xD4	GMAC_RXUDAR	31:24	RXUDA[31:24]							
		23:16	RXUDA[23:16]							
		15:8	RXUDA[15:8]							
		7:0	RXUDA[7:0]							
0xD8	GMAC_TXUDAR	31:24	TXUDA[31:24]							
		23:16	TXUDA[23:16]							
		15:8	TXUDA[15:8]							
		7:0	TXUDA[7:0]							
0xDC	GMAC_NSC	31:24								
		23:16			NANOSEC[21:16]					
		15:8	NANOSEC[15:8]							
		7:0	NANOSEC[7:0]							
0xE0	GMAC_SCL	31:24	SEC[31:24]							
		23:16	SEC[23:16]							
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							
0xE4	GMAC_SCH	31:24								
		23:16								
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							
0xE8	GMAC_EFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0xEC	GMAC_EFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0xF0	GMAC_PEFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0xF4	GMAC_PEFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0xF8 ... 0xFF	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100	GMAC_OTLO	31:24	TXO[31:24]							
		23:16	TXO[23:16]							
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x0104	GMAC_OTHI	31:24								
		23:16								
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x0108	GMAC_FT	31:24	FTX[31:24]							
		23:16	FTX[23:16]							
		15:8	FTX[15:8]							
		7:0	FTX[7:0]							
0x010C	GMAC_BCFT	31:24	BFTX[31:24]							
		23:16	BFTX[23:16]							
		15:8	BFTX[15:8]							
		7:0	BFTX[7:0]							
0x0110	GMAC_MFT	31:24	MFTX[31:24]							
		23:16	MFTX[23:16]							
		15:8	MFTX[15:8]							
		7:0	MFTX[7:0]							
0x0114	GMAC_PFT	31:24								
		23:16								
		15:8	PFTX[15:8]							
		7:0	PFTX[7:0]							
0x0118	GMAC_BFT64	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x011C	GMAC_TBFT127	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x0120	GMAC_TBFT255	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x0124	GMAC_TBFT511	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x0128	GMAC_TBFT1023	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x012C	GMAC_TBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x0130	GMAC_GTBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x0134	GMAC_TUR	31:24								
		23:16								
		15:8							TXUNR[9:8]	
		7:0	TXUNR[7:0]							
0x0138	GMAC_SCF	31:24								
		23:16							SCOL[17:16]	
		15:8	SCOL[15:8]							
		7:0	SCOL[7:0]							
0x013C	GMAC_MCF	31:24								
		23:16							MCOL[17:16]	
		15:8	MCOL[15:8]							
		7:0	MCOL[7:0]							
0x0140	GMAC_EC	31:24								
		23:16								
		15:8							XCOL[9:8]	
		7:0	XCOL[7:0]							
0x0144	GMAC_LC	31:24								
		23:16								
		15:8							LCOL[9:8]	
		7:0	LCOL[7:0]							
0x0148	GMAC_DTF	31:24								
		23:16							DEFT[17:16]	
		15:8	DEFT[15:8]							
		7:0	DEFT[7:0]							
0x014C	GMAC_CSE	31:24								
		23:16								
		15:8							CSR[9:8]	
		7:0	CSR[7:0]							
0x0150	GMAC_ORLO	31:24	RXO[31:24]							
		23:16	RXO[23:16]							
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							
0x0154	GMAC_ORHI	31:24								
		23:16								
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0158	GMAC_FR	31:24	FRX[31:24]							
		23:16	FRX[23:16]							
		15:8	FRX[15:8]							
		7:0	FRX[7:0]							
0x015C	GMAC_BCFR	31:24	BFRX[31:24]							
		23:16	BFRX[23:16]							
		15:8	BFRX[15:8]							
		7:0	BFRX[7:0]							
0x0160	GMAC_MFR	31:24	MFRX[31:24]							
		23:16	MFRX[23:16]							
		15:8	MFRX[15:8]							
		7:0	MFRX[7:0]							
0x0164	GMAC_PFR	31:24								
		23:16								
		15:8	PFRX[15:8]							
		7:0	PFRX[7:0]							
0x0168	GMAC_BFR64	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x016C	GMAC_TBFR127	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x0170	GMAC_TBFR255	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x0174	GMAC_TBFR511	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x0178	GMAC_TBFR1023	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x017C	GMAC_TBFR1518	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x0180	GMAC_TMXBFR	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0184	GMAC_UFR	31:24								
		23:16								
		15:8							UFRX[9:8]	
		7:0	UFRX[7:0]							
0x0188	GMAC_OFR	31:24								
		23:16								
		15:8							OFRX[9:8]	
		7:0	OFRX[7:0]							
0x018C	GMAC_JR	31:24								
		23:16								
		15:8							JRX[9:8]	
		7:0	JRX[7:0]							
0x0190	GMAC_FCSE	31:24								
		23:16								
		15:8							FCKR[9:8]	
		7:0	FCKR[7:0]							
0x0194	GMAC_LFFE	31:24								
		23:16								
		15:8							LFER[9:8]	
		7:0	LFER[7:0]							
0x0198	GMAC_RSE	31:24								
		23:16								
		15:8							RXSE[9:8]	
		7:0	RXSE[7:0]							
0x019C	GMAC_AE	31:24								
		23:16								
		15:8							AER[9:8]	
		7:0	AER[7:0]							
0x01A0	GMAC_RRE	31:24								
		23:16							RXRER[17:16]	
		15:8	RXRER[15:8]							
		7:0	RXRER[7:0]							
0x01A4	GMAC_ROE	31:24								
		23:16								
		15:8							RXOVR[9:8]	
		7:0	RXOVR[7:0]							
0x01A8	GMAC_IHCE	31:24								
		23:16								
		15:8								
		7:0	HCKER[7:0]							
0x01AC	GMAC_TCE	31:24								
		23:16								
		15:8								
		7:0	TCKER[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01B0	GMAC_UCE	31:24								
		23:16								
		15:8								
		7:0	UCKER[7:0]							
0x01B4	GMAC_FLRXPCR	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x01B8 ... 0x01BB	Reserved									
0x01BC	GMAC_TISUBN	31:24	LSBTIR[7:0]							
		23:16								
		15:8	MSBTIR[15:8]							
		7:0	MSBTIR[7:0]							
0x01C0	GMAC_TSH	31:24								
		23:16								
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x01C4 ... 0x01CF	Reserved									
0x01D0	GMAC_TSL	31:24	TCS[31:24]							
		23:16	TCS[23:16]							
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x01D4	GMAC_TN	31:24			TNS[29:24]					
		23:16	TNS[23:16]							
		15:8	TNS[15:8]							
		7:0	TNS[7:0]							
0x01D8	GMAC_TA	31:24	ADJ		ITDT[29:24]					
		23:16	ITDT[23:16]							
		15:8	ITDT[15:8]							
		7:0	ITDT[7:0]							
0x01DC	GMAC_TI	31:24								
		23:16	NIT[7:0]							
		15:8	ACNS[7:0]							
		7:0	CNS[7:0]							
0x01E0	GMAC_EFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01E4	GMAC_EFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01E8	GMAC_EFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01EC	GMAC_EFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01F0	GMAC_PEFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01F4	GMAC_PEFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01F8	GMAC_PEFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x01FC	GMAC_PEFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x0200 ... 0x025F	Reserved									
0x0260	GMAC_TXPQUANT1	31:24	QUANT_P3[15:8]							
		23:16	QUANT_P3[7:0]							
		15:8	QUANT_P2[15:8]							
		7:0	QUANT_P2[7:0]							
0x0264	GMAC_TXPQUANT2	31:24	QUANT_P5[15:8]							
		23:16	QUANT_P5[7:0]							
		15:8	QUANT_P4[15:8]							
		7:0	QUANT_P4[7:0]							
0x0268	GMAC_TXPQUANT3	31:24	QUANT_P7[15:8]							
		23:16	QUANT_P7[7:0]							
		15:8	QUANT_P6[15:8]							
		7:0	QUANT_P6[7:0]							
0x026C ... 0x026F	Reserved									
0x0270	GMAC_RXLPI	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0274	GMAC_RXLPITIME	31:24								
		23:16	LPITIME[23:16]							
		15:8	LPITIME[15:8]							
		7:0	LPITIME[7:0]							
0x0278	GMAC_TXLPI	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x027C	GMAC_TXLPITIME	31:24								
		23:16	LPITIME[23:16]							
		15:8	LPITIME[15:8]							
		7:0	LPITIME[7:0]							
0x0280 ... 0x02DF	Reserved									
0x02E0	GMAC_QOS_CFG0	31:24	Q3_DESCR[3:0]				Q3_DATA[3:0]			
		23:16	Q2_DESCR[3:0]				Q2_DATA[3:0]			
		15:8	Q1_DESCR[3:0]				Q1_DATA[3:0]			
		7:0	Q0_DESCR[3:0]				Q0_DATA[3:0]			
0x02E4	GMAC_QOS_CFG1	31:24								
		23:16								
		15:8	Q6_DESCR[3:0]				Q6_DATA[3:0]			
		7:0	Q5_DESCR[3:0]				Q5_DATA[3:0]			
0x02E8 ... 0x03FF	Reserved									
0x0400	GMAC_ISRPQ1	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0404	GMAC_ISRPQ2	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0408	GMAC_ISRPQ3	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x040C	GMAC_ISRPQ4	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0410	GMAC_ISRPQ5	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0414 ... 0x043F	Reserved									
0x0440	GMAC_TBQBAPQ1	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x0444	GMAC_TBQBAPQ2	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x0448	GMAC_TBQBAPQ3	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x044C	GMAC_TBQBAPQ4	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x0450	GMAC_TBQBAPQ5	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x0454 ... 0x047F	Reserved									
0x0480	GMAC_RBQBAPQ1	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x0484	GMAC_RBQBAPQ2	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x0488	GMAC_RBQBAPQ3	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x048C	GMAC_RBQBAPQ4	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x0490	GMAC_RBQBAPQ5	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0494 ... 0x049F	Reserved									
0x04A0	GMAC_RBSRPQ1	31:24								
		23:16								
		15:8								
		7:0	RBS[7:0]							
0x04A4	GMAC_RBSRPQ2	31:24								
		23:16								
		15:8								
		7:0	RBS[7:0]							
0x04A8	GMAC_RBSRPQ3	31:24								
		23:16								
		15:8								
		7:0	RBS[7:0]							
0x04AC	GMAC_RBSRPQ4	31:24								
		23:16								
		15:8								
		7:0	RBS[7:0]							
0x04B0	GMAC_RBSRPQ5	31:24								
		23:16								
		15:8								
		7:0	RBS[7:0]							
0x04B4 ... 0x04BB	Reserved									
0x04BC	GMAC_CBSCR	31:24								
		23:16								
		15:8								
		7:0							QBE	QAE
0x04C0	GMAC_CBSISQA	31:24	IS[31:24]							
		23:16	IS[23:16]							
		15:8	IS[15:8]							
		7:0	IS[7:0]							
0x04C4	GMAC_CBSISQB	31:24	IS[31:24]							
		23:16	IS[23:16]							
		15:8	IS[15:8]							
		7:0	IS[7:0]							
0x04C8	GMAC_TQUBA	31:24	TQUBA[31:24]							
		23:16	TQUBA[23:16]							
		15:8	TQUBA[15:8]							
		7:0	TQUBA[7:0]							
0x04CC	GMAC_TXBDCTRL	31:24								
		23:16								
		15:8								
		7:0			TSMODE[1:0]					

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04D0	GMAC_RXBDCTRL	31:24								
		23:16								
		15:8								
		7:0			TSMODE[1:0]					
0x04D4	GMAC_RQUBA	31:24	RQUBA[31:24]							
		23:16	RQUBA[23:16]							
		15:8	RQUBA[15:8]							
		7:0	RQUBA[7:0]							
0x04D8 ... 0x04FF	Reserved									
0x0500	GMAC_ST1RPQ0	31:24			UDPE	DSTCE	UDPM[15:12]			
		23:16	UDPM[11:4]							
		15:8	UDPM[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNB[2:0]		
0x0504	GMAC_ST1RPQ1	31:24			UDPE	DSTCE	UDPM[15:12]			
		23:16	UDPM[11:4]							
		15:8	UDPM[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNB[2:0]		
0x0508	GMAC_ST1RPQ2	31:24			UDPE	DSTCE	UDPM[15:12]			
		23:16	UDPM[11:4]							
		15:8	UDPM[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNB[2:0]		
0x050C	GMAC_ST1RPQ3	31:24			UDPE	DSTCE	UDPM[15:12]			
		23:16	UDPM[11:4]							
		15:8	UDPM[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNB[2:0]		
0x0510 ... 0x053F	Reserved									
0x0540	GMAC_ST2RPQ0	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x0544	GMAC_ST2RPQ1	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x0548	GMAC_ST2RPQ2	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x054C	GMAC_ST2RPQ3	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0550	GMAC_ST2RPQ4	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x0554	GMAC_ST2RPQ5	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x0558	GMAC_ST2RPQ6	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x055C	GMAC_ST2RPQ7	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x0560 ... 0x057F	Reserved									
0x0580	GMAC_TSCTL	31:24								
		23:16								
		15:8					TXSQ5[1:0]		TXSQ4[1:0]	
		7:0	TXSQ3[1:0]		TXSQ2[1:0]		TXSQ1[1:0]		TXSQ0[1:0]	
0x0584 ... 0x058F	Reserved									
0x0590	GMAC_TQBWRL0	31:24	ALLOCQ3[7:0]							
		23:16	ALLOCQ2[7:0]							
		15:8	ALLOCQ1[7:0]							
		7:0	ALLOCQ0[7:0]							
0x0594	GMAC_TQBWRL1	31:24								
		23:16								
		15:8	ALLOCQ5[7:0]							
		7:0	ALLOCQ4[7:0]							
0x0598 ... 0x059F	Reserved									
0x05A0	GMAC_TQSA	31:24								
		23:16		SEGALLOCQ5[2:0]				SEGALLOCQ4[2:0]		
		15:8		SEGALLOCQ3[2:0]				SEGALLOCQ2[2:0]		
		7:0		SEGALLOCQ1[2:0]				SEGALLOCQ0[2:0]		
0x05A4 ... 0x05FF	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0600	GMAC_IERPQ1	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0604	GMAC_IERPQ2	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0608	GMAC_IERPQ3	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x060C	GMAC_IERPQ4	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0610	GMAC_IERPQ5	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0614 ... 0x061F	Reserved									
0x0620	GMAC_IDRPQ1	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0624	GMAC_IDRPQ2	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0628	GMAC_IDRPQ3	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x062C	GMAC_IDRPQ4	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0630	GMAC_IDRPQ5	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
0x0634 ... 0x063F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0640	GMAC_IMRPQ1	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	AHB	RLEX			RXUBR	RCOMP	
0x0644	GMAC_IMRPQ2	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	AHB	RLEX			RXUBR	RCOMP	
0x0648	GMAC_IMRPQ3	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	AHB	RLEX			RXUBR	RCOMP	
0x064C	GMAC_IMRPQ4	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	AHB	RLEX			RXUBR	RCOMP	
0x0650	GMAC_IMRPQ5	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	AHB	RLEX			RXUBR	RCOMP	
0x0654 ... 0x06DF	Reserved									
0x06E0	GMAC_ST2ER0	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x06E4	GMAC_ST2ER1	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x06E8	GMAC_ST2ER2	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x06EC	GMAC_ST2ER3	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x06F0 ... 0x06FF	Reserved									
0x0700	GMAC_ST2CW0R0	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0704	GMAC_ST2CW1R0	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0708	GMAC_ST2CW0R1	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x070C	GMAC_ST2CW1R1	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0710	GMAC_ST2CW0R2	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0714	GMAC_ST2CW1R2	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0718	GMAC_ST2CW0R3	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x071C	GMAC_ST2CW1R3	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0720	GMAC_ST2CW0R4	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0724	GMAC_ST2CW1R4	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0728	GMAC_ST2CW0R5	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x072C	GMAC_ST2CW1R5	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0730	GMAC_ST2CW0R6	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0734	GMAC_ST2CW1R6	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0738	GMAC_ST2CW0R7	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x073C	GMAC_ST2CW1R7	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0740	GMAC_ST2CW0R8	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0744	GMAC_ST2CW1R8	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0748	GMAC_ST2CW0R9	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x074C	GMAC_ST2CW1R9	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0750	GMAC_ST2CW0R10	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0754	GMAC_ST2CW1R10	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0758	GMAC_ST2CW0R11	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x075C	GMAC_ST2CW1R11	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0760	GMAC_ST2CW0R12	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0764	GMAC_ST2CW1R12	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0768	GMAC_ST2CW0R13	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x076C	GMAC_ST2CW1R13	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0770	GMAC_ST2CW0R14	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0774	GMAC_ST2CW1R14	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0778	GMAC_ST2CW0R15	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x077C	GMAC_ST2CW1R15	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0780	GMAC_ST2CW0R16	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0784	GMAC_ST2CW1R16	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0788	GMAC_ST2CW0R17	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x078C	GMAC_ST2CW1R17	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0790	GMAC_ST2CW0R18	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x0794	GMAC_ST2CW1R18	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x0798	GMAC_ST2CW0R19	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x079C	GMAC_ST2CW1R19	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x07A0	GMAC_ST2CW0R20	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x07A4	GMAC_ST2CW1R20	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x07A8	GMAC_ST2CW0R21	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x07AC	GMAC_ST2CW1R21	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x07B0	GMAC_ST2CW0R22	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x07B4	GMAC_ST2CW1R22	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x07B8	GMAC_ST2CW0R23	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x07BC	GMAC_ST2CW1R23	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						

62.8.1 GMAC Network Control Register

Name: GMAC_NCR
Offset: 0x000
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		IFGQAVCRED		MIIONRGMII	OSSCORR	EXTSELRQEN	PFCCTL	OSSMODE
Access		R/W		R/W	R/W	R/W	R/W	R/W
Reset		0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		STUDPOFFSET		PTPUNIENA	TXLPIEN	FNP	TXPBPF	ENPBPR
Access		R/W		R/W	R/W	R/W	R/W	R/W
Reset		0		0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
Access	R/W	W	W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit 30 – IFGQAVCRED Credit-Based Shaping Algorithm Modification

Value	Description
0	No modification of the CBS algorithm.
1	Modifies the CBS algorithm so the IFG/IPG associated with a transmit frame counts towards its 802.1Qav credit.

Bit 28 – MIIONRGMII MII Control

Value	Description
0	Disables MII operation.
1	Enables MII operation when GMAC_UR.MIM=0.

Bit 27 – OSSCORR OSS Correction Field
1588 One Step Correction field update. Set this bit high to update the correction field of PTP 1588 version 2 sync frames by adding current TSU timer value.

Bit 26 – EXTSELRQEN External Selection of Receive Queue Enable

Value	Description
0	Disables external selection of receive queue.
1	Enables external selection of receive queue.

Bit 25 – PFCCTL Multiple PFC Pause Quantum Enable

Value	Description
0	Disables multiple PFC pause quantums.
1	Enables multiple PFC pause quantums, one per pause priority.

Bit 24 – OSSMODE One Step Sync Mode

Value	Description
0	1588 One Step Sync mode is disabled.

Value	Description
1	1588 One Step Sync mode is enabled. Replaces timestamp field in the 1588 header for TX Sync Frames with the current TSU timer value.

Bit 22 – STUDPOFFSET Store UDP Offset
Stores UDP/TCP offset to memory.

Value	Description
0	Normal operations.
1	The upper 16 bits of the CRC of every received frame are replaced with the offset from start of frame to the beginning of the UDP or TCP header. The lower 16 bits of the CRC are replaced with zero and reserved for future use. The offset is measured in units of 2 bytes.

Bit 20 – PTPUNIENA Detection of Unicast PTP Frames Enable

Value	Description
0	Disables detection of unicast PTP frames.
1	Enables detection of unicast PTP frames.

Bit 19 – TXLPIEN Enable LPI Transmission
When set, LPI (low power idle) is immediately transmitted.

Bit 18 – FNP Flush Next Packet
Flush the next packet from the external receive memory. Writing one to this bit will only have an effect if the DMA is not currently writing a packet already stored in the receive memory to system memory.

Value	Description
0	No effect.
1	Flushes the next packet from the receive memory. This will only have an effect if the DMA is not currently writing a packet already stored in the receive memory to system memory.

Bit 17 – TXPBPF Transmit PFC Priority-based Pause Frame

Value	Description
0	No effect.
1	Takes the values stored in the Transmit PFC Pause Register.

Bit 16 – ENPBPR Enable PFC Priority-based Pause Reception
Enables PFC Priority Based Pause Reception capabilities. Setting this bit enables PFC negotiation and recognition of priority-based pause frames.

Bit 15 – SRTSM Store Receive Timestamp to Memory

Value	Description
0	No effect.
1	Causes the CRC of every received frame to be replaced with the value of the nanoseconds field of the 1588 timer that was captured as the receive frame passed the message timestamp point. Note that bit RFCS in register GMAC_NCFGR may not be set to 1 when the timer should be captured.

Bit 12 – TXZQPF Transmit Zero Quantum Pause Frame

Value	Description
0	No effect.
1	Generates a pause frame with zero quantum to be transmitted.

Bit 11 – TXPF Transmit Pause Frame

Value	Description
0	No effect.
1	Generates a pause frame to be transmitted.

Bit 10 – THALT Transmit Halt

Value	Description
0	No effect.

Value	Description
1	Halts transmission as soon as any ongoing frame transmission ends.

Bit 9 – TSTART Start Transmission

Value	Description
0	No effect.
1	Starts transmission.

Bit 8 – BP Back pressure

Value	Description
0	No effect.
1	When the MAC is set in 10M or 100M Half Duplex mode, forces collisions on all received frames. Ignored in Gigabit Half Duplex mode.

Bit 7 – WESTAT Write Enable for Statistics Registers

Value	Description
0	Forces the statistics registers to be in read-only mode for normal operation mode.
1	Makes the statistics registers writable for functional test purposes.

Bit 6 – INCSTAT Increment Statistics Registers

Bit 5 – CLRSTAT Clear Statistics Registers

Value	Description
0	No effect.
1	Clears the statistics registers.

Bit 4 – MPE Management Port Enable
Set to one to enable the management port. When zero, forces GMDIO to high impedance state and MDC low.

Value	Description
0	Forces GMDIO to high impedance state and MDC low.
1	Enables the management port.

Bit 3 – TXEN Transmit Enable

Value	Description
0	Stops transmission immediately, the transmit pipeline and control registers will be cleared and the Transmit Queue Pointer register will reset to point to the start of the transmit descriptor list.
1	Enables the GMAC transmitter to send data.

Bit 2 – RXEN Receive Enable
When set, RXEN enables the GMAC to receive data. When reset frame reception stops immediately and the receive pipeline will be cleared. The Receive Queue Pointer Register is unaffected.

Value	Description
0	Stops frame reception immediately and the receive pipeline will be cleared. The Receive Queue Pointer register is unaffected.
1	Enables the GMAC to receive data.

Bit 1 – LBL Loop Back Local

Value	Description
0	Normal operating mode (no loop back).
1	Connects GTX to GRX, GTXEN to GRXDV and forces Full Duplex mode. GRXCK and GTXCK may malfunction as the GMAC is switched into and out of internal loop back. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loop back.

62.8.2 GMAC Network Configuration Register

Name: GMAC_NCFGR
Offset: 0x004
Reset: 0x00080000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0
Bit	23	22	21	20	19	18	17	16
	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXBUFO[1:0]		PEN	RTY		GBE		MAXFS
Access	R/W	R/W	R/W	R/W		R/W		R/W
Reset	0	0	0	0		0		0
Bit	7	6	5	4	3	2	1	0
	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – IRXER Ignore Receive Error from PHY
When set, GRXER has no effect on the GMAC's operation when GRXDV is low. Set this bit when using the RGMII wrapper in Half Duplex mode.

Bit 29 – RXBP Receive Bad Preamble
When set, frames with non-standard preamble are not rejected.

Bit 28 – IPGSEN IP Stretch Enable
When set, the transmit IPG can be increased above 96 bit times depending on the previous frame length using the IPG Stretch Register.

Bit 26 – IRXFCS Ignore RX FCS
When set, frames with FCS/CRC errors will not be rejected. FCS error statistics will still be collected for frames with bad FCS and FCS status will be recorded in frame’s DMA descriptor. For normal operation this bit must be set to zero.

Bit 25 – EFRHD Enable Frames Received in Half Duplex
Enable frames to be received in half-duplex mode while transmitting.

Bit 24 – RXCOEN Receive Checksum Offload Enable
When set, the receive checksum engine is enabled. Frames with bad IP, TCP or UDP checksums are discarded.

Bit 23 – DCPF Disable Copy of Pause Frames
Set to one to prevent valid pause frames being copied to memory. When set, pause frames are not copied to memory regardless of the state of the Copy All Frames bit, whether a hash match is found or whether a type ID match is identified. If a destination address match is found, the pause frame will be copied to memory. Note that valid pause frames received will still increment pause statistics and pause the transmission of frames as required.

Bits 22:21 – DBW[1:0] Data Bus Width
The default value for this register is 64 bits. Must always be written to ‘1’.

Bits 20:18 – CLK[2:0] MDC CLock Division
Set according to MCK speed. These three bits determine the number MCK will be divided by to generate Management Data Clock (MDC). For conformance with the 802.3 specification, MDC must not exceed 2.5 MHz (MDC is only active during MDIO read and write operations).

Value	Name	Description
0	MCK_8	MCK divided by 8 (MCK up to 20 MHz)
1	MCK_16	MCK divided by 16 (MCK up to 40 MHz)
2	MCK_32	MCK divided by 32 (MCK up to 80 MHz)
3	MCK_48	MCK divided by 48 (MCK up to 120 MHz)
4	MCK_64	MCK divided by 64 (MCK up to 160 MHz)
5	MCK_96	MCK divided by 96 (MCK up to 240 MHz)

Bit 17 – RFCS Remove FCS
Setting this bit will cause received frames to be written to memory without their frame check sequence (last 4 bytes). The frame length indicated will be reduced by four bytes in this mode.

Bit 16 – LFERD Length Field Error Frame Discard
Setting this bit causes frames with a measured length shorter than the extracted length field (as indicated by bytes 13 and 14 in a non-VLAN tagged frame) to be discarded. This only applies to frames with a length field less than 0x0600.

Bits 15:14 – RXBUFO[1:0] Receive Buffer Offset
Indicates the number of bytes by which the received data is offset from the start of the receive buffer

Bit 13 – PEN Pause Enable
When set, transmission will pause if a non-zero 802.3 classic pause frame is received and PFC has not been negotiated.

Bit 12 – RTY Retry Test
Must be set to zero for normal operation. If set to one the backoff between collisions will always be one slot time. Setting this bit to one helps test the too many retries condition. Also used in the pause frame tests to reduce the pause counter's decrement time from 512 bit times, to every GRXCK cycle.

Bit 10 – GBE Gigabit Mode Enable
Setting this bit configures the GMAC for 1000 Mbps operation.

Value	Description
0	10/100 operation.
1	Gigabit operation.

Bit 8 – MAXFS 1536 Maximum Frame Size
Setting this bit means the GMAC will accept frames up to 1536 bytes in length. Normally the GMAC would reject any frame above 1518 bytes.

Bit 7 – UNIHEN Unicast Hash Enable
When set, unicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.

Bit 6 – MTIHEN Multicast Hash Enable
When set, multicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.

Bit 5 – NBC No Broadcast
When set to logic one, frames addressed to the broadcast address of all ones will not be accepted.

Bit 4 – CAF Copy All Frames
When set to logic one, all valid frames will be accepted.

Bit 3 – JFRAME Jumbo Frame Size
Set to one to enable jumbo frames up to 10240 bytes to be accepted. The default length is 10240 bytes.

Bit 2 – DNVLAN Discard Non-VLAN FRAMES
When set only VLAN tagged frames will be passed to the address matching logic.

Bit 1 – FD Full Duplex
If set to logic one, the transmit block ignores the state of collision and carrier sense and allows receive while transmitting.

Bit 0 – SPD Speed
Set to logic one to indicate 100 Mbps operation, logic zero for 10 Mbps.

62.8.3 GMAC Network Status Register

Name: GMAC_NSR
Offset: 0x008
Reset: see Note
Property: Read-only

Note: The register reset value is either 0x00000004 or 0x00000006 depending on the status of the GMDIO input pin.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SB_PEND_XFE R
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	RXLPIS	PFCPAUSN				IDLE	MDIO	
Access	R	R				R	R	
Reset	0	0				1	x	

Bit 8 – SB_PEND_XFER System Bus Pending Transactions
Set when read or write transactions have been issued on system bus but the responses have not yet been collected.

Bit 7 – RXLPIS LPI Indication
Low power idle has been detected on receive. This bit is set when LPI is detected and reset when normal idle is detected. An interrupt is generated when the state of this bit changes.

Bit 6 – PFCPAUSN PFC Pause Negotiated
Set when PFC Priority-based Pause has been negotiated.

Bit 2 – IDLE PHY Management Logic Idle
The PHY management logic is idle (i.e., has completed).

Bit 1 – MDIO MDIO Input Status
Returns status of the GMDIO pin.

62.8.4 GMAC User Register

Name: GMAC_UR
Offset: 0x00C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		HDFLCTLEN				REFCLK	MIM[1:0]	
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 6 – HDFLCTLEN Half Duplex Flow Control Enable

Value	Description
0	Half duplex flow control is disabled.
1	Half duplex flow control is enabled.

Bit 2 – REFCLK Source for the GMAC Reference Clock

Value	Name	Description
0	INTERNAL_GCLK	Selects the GCLK from PMC.
1	EXTERNAL	Selects the clock from an IO.

Bits 1:0 – MIM[1:0] Media Interface Mode

Value	Name	Description
0	MII	Selects MII mode when GMAC_NCR.MIIONRGMII=1.
1	RMII	Selects RMII mode when GMAC_NCR.MIIONRGMII=0.
2	RGMII	Selects RGMII mode when GMAC_NCR.MIIONRGMII=0.

62.8.5 GMAC DMA Configuration Register

Name: GMAC_DCFGR
Offset: 0x010
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			TXBD_EXTENDED	RXBD_EXTENDED				DDRP
Access			R/W	R/W				R/W
Reset			0	0				0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	DRBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
				INFLASTEN	TXCOEN	TXPBMS	RXBMS[1:0]	
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	ESPA	ESMA		FBLDO[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 29 – TXBD_EXTENDED Transmit Buffer Descriptor Extended Mode
See GMAC Transmit Buffer Data Control Register for a description of the features.

Value	Description
0	Disables Transmit Buffer Data Extended mode.
1	Enables Transmit Buffer Data Extended mode.

Bit 28 – RXBD_EXTENDED Receive Buffer Descriptor Extended Mode
See GMAC Receive Buffer Data Control Register for a description of the features.

Value	Description
0	Disables Receive Buffer Data Extended mode.
1	Enables Receive Buffer Data Extended mode.

Bit 24 – DDRP DMA Discard Receive Packets
When set, the GMAC DMA automatically discards receive packets from the receiver packet buffer memory when no system memory resource is available.
When low, the received packets remain to be stored in the GMAC local memory packet buffer until a system memory buffer resource becomes available.
A write to this bit is ignored if the DMA is not configured in the packet buffer full store and forward mode.

Bits 23:16 – DRBS[7:0] DMA Receive Buffer Size
DMA receive buffer size in system memory. The value defined by these bits determines the size of buffer to use in main system memory when writing received data.
The value is defined in multiples of 64 bytes, thus a value of 0x01 corresponds to buffers of 64 bytes, 0x02 corresponds to 128 bytes etc.
For example:
– 0x02: 128 bytes
– 0x18: 1536 bytes (1 × max length frame/buffer)
– 0xA0: 10240 bytes (1 × 10K jumbo frame/buffer)

Note that this value should never be written as zero.

Bit 12 – INFLASTEN Infinite Size for Last Buffer Enable
Set to one, this forces the receive DMA to consider the data buffer pointed to by the last descriptor in the descriptor list to be of definite size.

Bit 12 – CRCERRREP CRC Errors Report

Value	Description
0	Bit 16 of the receive buffer descriptor represents the Canonical format indicator (CFI) bit as extracted from the receive frame (if the receive buffer descriptor is pointing to the last data buffer of the receive frame and the received frame was VLAN tagged).
1	Bit 16 of the receive buffer descriptor represents the FCS/CRC error (only if frames with FCS are copied to memory as enabled by bit 26 in the Network Configuration register).

Bit 11 – TXCOEN Transmitter Checksum Generation Offload Enable
Transmitter IP, TCP and UDP checksum generation offload enable. When set, the transmitter checksum generation engine is enabled to calculate and substitute checksums for transmit frames. When clear, frame data is unaffected.

Bit 10 – TXPBMS Transmitter Packet Buffer Memory Size Select
Having this bit at zero halves the amount of memory used for the transmit packet buffer. This reduces the amount of memory used by the GMAC. It is important to set this bit to one if the full configured physical memory is available. The value in brackets below represents the size that would result for the default maximum configured memory size of 4 Kbytes.

Value	Name	Description
0	TWO_KB	Do not use top address bit (2 Kbytes).
1	FOUR_KB	Use full configured addressable space (4 Kbytes).

Bits 9:8 – RXBMS[1:0] Receiver Packet Buffer Memory Size Select
The default receive packet buffer size is 8 Kbytes. The table below shows how to configure this memory to FULL, HALF, QUARTER or EIGHTH of the default size.

Value	Name	Description
0	EIGHTH	8/8 Kbyte memory size
1	QUARTER	8/4 Kbytes memory size
2	HALF	8/2 Kbytes memory size
3	FULL	8 Kbytes memory size

Bit 7 – ESPA Endian Swap Mode Enable for Packet Data Accesses
When set, selects swapped endianism for system bus transfers. When clear, selects Little Endian mode.

Value	Name	Description
0	LITTLE_ENDIAN	Selects Little-endian endianism for system bus transfers.
1	BIG_ENDIAN	Selects swapped endianism for system bus transfers.

Bit 6 – ESMA Endian Swap Mode Enable for Management Descriptor Accesses
When set, selects swapped endianism for system bus transfers. When clear, selects Little Endian mode.

Value	Name	Description
0	LITTLE_ENDIAN	Selects Little-endian endianism for system bus transfers.
1	BIG_ENDIAN	Selects swapped endianism for system bus transfers.

Bits 4:0 – FBLDO[4:0] Fixed Burst Length for DMA Data Operations
Selects the burst length to attempt to use on the system bus when transferring frame data. Not used for DMA management operations and only used where space and data size allow. Otherwise system bus single type accesses are used.

Value	Name	Description
0	–	Reserved
1	SINGLE	Always use single access on system bus
2	–	Reserved
4	INCR4	Attempt to use 4-beat bursts on system bus (Default)

Value	Name	Description
8	INCR8	Attempt to use 8-beat bursts on system bus bursts
16	INCR16	Attempt to use 16-beat bursts on system bus bursts

62.8.6 GMAC Transmit Status Register

Name: GMAC_TSR
Offset: 0x014
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								HRESP
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	LCO		TXCOMP	TFC	TXGO	RLE	COL	UBR
Access	R/W		R/W	R/W	R	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 8 – HRESP System Bus Response
Set when the DMA block sees a system bus error. Writing a one clears this bit.

Bit 7 – LCO Late Collision Occurred
Only set if the condition occurs in Gigabit mode, as retry is not attempted. Writing a one clears this bit.

Bit 5 – TXCOMP Transmit Complete
Set when a frame has been transmitted. Writing a one clears this bit.

Bit 4 – TFC Transmit Frame Corruption Due to System Bus Error
Transmit frame corruption due to system bus error. Set if an error occurs while midway through reading transmit frame from the system bus, including system bus errors and buffers exhausted mid frame (if the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and GTXER asserted).
Also set in DMA packet buffer mode if single frame is too large for configured packet buffer memory size.
Writing a one clears this bit.

Bit 3 – TXGO Transmit Go (Read only)
When high, transmit is active. When using the DMA interface, this bit represents the TXGO variable as specified in the transmit buffer description.

Bit 2 – RLE Retry Limit Exceeded
Writing a one clears this bit.

Bit 1 – COL Collision Occurred
Set by the assertion of collision. Writing a one clears this bit. When operating in 10/100 mode, this status indicates either a collision or a late collision. In gigabit mode, this status is not set for a late collision.

Bit 0 – UBR Used Bit Read
Set when a transmit buffer descriptor is read with its used bit set. Writing a one clears this bit.

62.8.7 GMAC Receive Buffer Queue Base Address Register

Name: GMAC_RBQB
Offset: 0x018
Reset: 0x00000000
Property: Read/Write

This register holds the start address of the receive buffer queue (receive buffers descriptor list). The receive buffer queue base address must be initialized before receive is enabled through bit 2 of the Network Control Register. Once reception is enabled, any write to the Receive Buffer Queue Base Address Register is ignored. Reading this register returns the location of the descriptor currently being accessed. This value increments as buffers are used. Software should not use this register for determining where to remove received frames from the queue as it constantly changes as new frames are received. Software should instead work its way through the buffer descriptor queue checking the “used” bits.

When the datapath is configured at 64 bits, the descriptors must be aligned at 64-bit boundaries and each pair of 32-bit descriptors is written to by using a single access. The descriptors must be aligned at 32-bit boundaries and the descriptors are written to using two individual non sequential accesses for 32-bit datapaths.

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]							RXQDIS
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bits 31:2 – ADDR[29:0] Receive Buffer Queue Base Address
Written with the address of the start of the receive queue.

Bit 0 – RXQDIS Receive Queue Disable

Value	Description
0	Queue is enabled.
1	Queue is disabled. Used to reduce the number of active queues and should only be changed while receive is not enabled.

62.8.8 GMAC Transmit Buffer Queue Base Address Register

Name: GMAC_TBQB
Offset: 0x01C
Reset: 0x00000000
Property: Read/Write

This register holds the start address of the transmit buffer queue (transmit buffers descriptor list). The Transmit Buffer Queue Base Address Register must be initialized before transmit is started through bit 9 of the Network Control Register. Once transmission has started, any write to the Transmit Buffer Queue Base Address Register is illegal and therefore ignored.

Note that due to clock boundary synchronization, it takes a maximum of four MCK cycles from the writing of the transmit start bit before the transmitter is active. Writing to the Transmit Buffer Queue Base Address Register during this time may produce unpredictable results.

Reading this register returns the location of the descriptor currently being accessed. Since the DMA handles two frames at once, this may not necessarily be pointing to the current frame being transmitted.

When the datapath is configured at 64 bits, the descriptors must be aligned at 64-bit boundaries and each pair of 32-bit descriptors is read from memory using a single access. The descriptors must be aligned at 32-bit boundaries and the descriptors are read from memory using two individual non sequential accesses for 32-bit datapaths.

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[5:0]							TXQDIS
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bits 31:2 – ADDR[29:0] Transmit Buffer Queue Base Address
Written with the address of the start of the transmit queue.

Bit 0 – TXQDIS Transmit Queue Disable

Value	Description
0	Queue is enabled.
1	Queue is disabled. Used to reduce the number of active queues and should only be changed while transmit is not enabled.

62.8.9 GMAC Receive Status Register

Name: GMAC_RSR
Offset: 0x020
Reset: 0x00000000
Property: Read/Write

This register, when read, provides receive status details. Once read, individual bits may be cleared by writing a one to them. It is not possible to set a bit to 1 by writing to the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					HNO	RXOVR	REC	BNA
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – HNO System Bus Error
Set when the DMA block sees a system bus error. Writing a one clears this bit.

Bit 2 – RXOVR Receive Overrun
This bit is set if the receive status was not taken at the end of the frame. This bit is also set if the packet buffer overflows. The buffer will be recovered if an overrun occurs. Writing a one clears this bit.

Bit 1 – REC Frame Received
One or more frames have been received and placed in memory. Writing a one clears this bit.

Bit 0 – BNA Buffer Not Available
An attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA will re-read the pointer each time an end of frame is received until a valid pointer is found. This bit is set following each descriptor read attempt that fails, even if consecutive pointers are unsuccessful and software has in the mean time cleared the status flag. Writing a one clears this bit.

62.8.10 GMAC Interrupt Status Register

Name: GMAC_ISR
Offset: 0x024
Reset: 0x00000000
Property: Read-only

This register indicates the source of the interrupt. In order that the bits of this register read 1, the corresponding interrupt source must be enabled in the mask register. If any bit is set in this register, the GMAC interrupt signal will be asserted in the system.

Bit	31	30	29	28	27	26	25	24
			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8
		PFTR	PTZ	PFNZ	HRESP	ROVR		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 29 – TSUTIMCOMP TSU Timer Comparison (cleared on read)
Indicates when the TSU timer count value is equal to programmed value.

Bit 28 – WOL Wake On LAN
WOL interrupt. Indicates a WOL event has been received.

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change (cleared on read)
Receive LPI indication status bit change.

Bit 26 – SRI TSU Seconds Register Increment (cleared on read)
Indicates the register has incremented.

Bit 25 – PDRSFT PDelay Response Frame Transmitted (cleared on read)
Indicates a PTP pdelay_resp frame has been transmitted.

Bit 24 – PDRQFT PDelay Request Frame Transmitted (cleared on read)
Indicates a PTP pdelay_req frame has been transmitted.

Bit 23 – PDRSFR PDelay Response Frame Received (cleared on read)
Indicates a PTP pdelay_resp frame has been received.

Bit 22 – PDRQFR PDelay Request Frame Received
Indicates a PTP pdelay_req frame has been received.

Bit 21 – SFT PTP Sync Frame Transmitted (cleared on read)
Indicates a PTP sync frame has been transmitted.

Bit 20 – DRQFT PTP Delay Request Frame Transmitted (cleared on read)
Indicates a PTP delay_req frame has been transmitted. (cleared on read)

Bit 19 – SFR PTP Sync Frame Received (cleared on read)
Indicates a PTP sync frame has been received.

Bit 18 – DRQFR PTP Delay Request Frame Received (cleared on read)
Indicates a PTP delay_req frame has been received.

Bit 14 – PFTR Pause Frame Transmitted (cleared on read)
Indicates a pause frame has been successfully transmitted after being initiated from the Network Control register.

Bit 13 – PTZ Pause Time Zero (cleared on read)
Set when either the Pause Time register at address 0x38 decrements to zero, or when a valid pause frame is received with a zero pause quantum field.

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received (cleared on read)
Indicates a valid pause has been received that has a non-zero pause quantum field.

Bit 11 – HRESP System Bus Error (cleared on read)
Set when the DMA block sees a system bus error.

Bit 10 – ROVR Receive Overrun (cleared on read)
Set when the receive overrun status bit is set.

Bit 7 – TCOMP Transmit Complete (cleared on read)
Set when a frame has been transmitted.

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error (cleared on read)
Set if an error occurs while midway through reading transmit frame from the system bus, including system bus error and buffers exhausted mid frame.

Bit 5 – RLEX Retry Limit Exceeded or Late Collision (cleared on read)
Transmit error. Late collision will only cause this status bit to be set in Gigabit mode, as a retry is not attempted.

Bit 4 – TUR Transmit Underrun (cleared on read)
This interrupt is set if the transmitter was forced to terminate a frame that it has already began transmitting due to further data being unavailable.
This interrupt is set if a transmitter status write back has not completed when another status write back is attempted.
This interrupt is also set when the transmit DMA has written the SOP data into the FIFO and either the system bus was not granted in time for further data, or because a system bus error response was returned, or because the used bit was read.

Bit 3 – TXUBR TX Used Bit Read (cleared on read)
Set when a transmit buffer descriptor is read with its used bit set.

Bit 2 – RXUBR RX Used Bit Read (cleared on read)
Set when a receive buffer descriptor is read with its used bit set.

Bit 1 – RCOMP Receive Complete (cleared on read)
A frame has been stored in memory.

Bit 0 – MFS Management Frame Sent (cleared on read)
The PHY Maintenance Register has completed its operation.

62.8.11 GMAC Interrupt Enable Register

Name: GMAC_IER
Offset: 0x028
Reset: –
Property: Write-only

This register is write-only and when read will return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – TSUTIMCOMP TSU Timer Comparison (cleared on read)
Indicates when the TSU timer count value is equal to programmed value.

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Enable RX LPI Indication

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP System Bus Error

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

62.8.12 GMAC Interrupt Disable Register

Name: GMAC_IDR
Offset: 0x02C
Reset: –
Property: Write-only

This register is write-only and when read will return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – TSUTIMCOMP TSU Timer Comparison (cleared on read)
Indicates when the TSU timer count value is equal to programmed value.

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Enable RX LPI Indication

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP System Bus Error

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

62.8.13 GMAC Interrupt Mask Register

Name: GMAC_IMR
Offset: 0x030
Reset: 0x07FFFFFF
Property: Read/Write

The Interrupt Mask Register is a read-only register indicating which interrupts are masked. All bits are set at reset and can be reset individually by writing to the Interrupt Enable Register or set individually by writing to the Interrupt Disable Register. Having separate address locations for enable and disable saves the need for performing a read modify write when updating the Interrupt Mask Register.

For test purposes there is a write-only function to this register that allows the bits in the Interrupt Status Register to be set or cleared, regardless of the state of the mask register. A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

The following values are valid for all listed bit names of this register when read:

0: The corresponding interrupt is enabled.

1: The corresponding interrupt is not enabled.

Bit	31	30	29	28	27	26	25	24
			TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	1	1
Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	1	1	1	1	1	1		
Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	1	1	1	1	1	1		
Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 29 – TSUTIMCOMP TSU Timer Comparison (cleared on read)
Indicates when the TSU timer count value is equal to programmed value.

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Enable RX LPI Indication

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP System Bus Error

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

62.8.14 GMAC PHY Maintenance Register

Name: GMAC_MAN
Offset: 0x034
Reset: 0x00000000
Property: Read/Write

The PHY Maintenance Register is implemented as a shift register. Writing to the register starts a shift operation which is signalled as complete when bit 2 is set in the Network Status Register. It takes about 2000 MCK cycles to complete, when MDC is set for MCK divide by 32 in the Network Configuration Register. An interrupt is generated upon completion.

During this time, the MSB of the register is output on the GMDIO pin and the LSB updated from the GMDIO pin with each MDC cycle. This causes transmission of a PHY management frame on the GMDIO pin. See Section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation returns the current contents of the shift register. At the end of management operation, the bits will have shifted back to their original locations. For a read operation, the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The MDIO interface can read IEEE 802.3 clause 45 PHYs as well as clause 22 PHYs. To read clause 45 PHYs, bit 30 should be written with a 0 rather than a 1. To write clause 45 PHYs, bits 31:28 should be written as 0x0001. See the table below.

Table 62-16. Clause 22/Clause 45 PHYs Read/Write Access Configuration (GMAC_MAN Bits 31:28)

PHY	Access	Bit Value			
		WZO	CLTTO	OP[1]	OP[0]
Clause 22	Read	0	1	1	0
	Write	0	1	0	1
Clause 45	Read	0	0	1	1
	Write	0	0	0	1
	Read + Address	0	0	1	0

For a description of MDC generation, see GMAC Network Configuration Register.

Bit	31	30	29	28	27	26	25	24
	WZO	CLTTO	OP[1:0]		PHYA[4:1]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PHYA[0]	REGA[4:0]					WTN[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – WZO Write ZERO
Must be written with 0.

Bit 30 – CLTTO Clause 22 Operation

Value	Description
0	Clause 45 operation
1	Clause 22 operation

Bits 29:28 – OP[1:0] Operation

Value	Description
01	Write
10	Read

Bits 27:23 – PHYA[4:0] PHY Address

Bits 22:18 – REGA[4:0] Register Address
Specifies the register in the PHY to access.

Bits 17:16 – WTN[1:0] Write Ten
Must be written to 10.

Bits 15:0 – DATA[15:0] PHY Data
For a write operation this field is written with the data to be written to the PHY. After a read operation this field contains the data read from the PHY.

62.8.15 GMAC Receive Pause Quantum Register

Name: GMAC_RPQ
Offset: 0x038
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RPQ[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RPQ[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RPQ[15:0] Received Pause Quantum
Stores the current value of the Receive Pause Quantum Register which is decremented every 512 bit times.

62.8.16 GMAC Transmit Pause Quantum Register

Name: GMAC_TPQ
Offset: 0x03C
Reset: 0xFFFFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	P1TPQ[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	P1TPQ[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	TPQ[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	TPQ[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:16 – P1TPQ[15:0] Priority 1 Transmit Pause Quantum
Written with the pause quantum value for pause frame transmission.

Bits 15:0 – TPQ[15:0] Transmit Pause Quantum
Written with the pause quantum value for pause frame transmission.

62.8.17 GMAC TX Partial Store and Forward Register

Name: GMAC_TPSF
Offset: 0x040
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENTXP							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						TPB1ADR[10:8]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	7	6	5	4	3	2	1	0
	TPB1ADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – ENTXP Enable TX Partial Store and Forward Operation

Bits 10:0 – TPB1ADR[10:0] Transmit Partial Store and Forward Address
Watermark value. Reset = 1.

62.8.18 GMAC RX Partial Store and Forward Register

Name: GMAC_RPSF
Offset: 0x044
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENRXP							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						RPB1ADR[10:8]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	7	6	5	4	3	2	1	0
	RPB1ADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – ENRXP Enable RX Partial Store and Forward Operation

Bits 10:0 – RPB1ADR[10:0] Receive Partial Store and Forward Address
Watermark value. Reset = 1.

62.8.19 GMAC RX Jumbo Frame Max Length Register

Name: GMAC_RJFML
Offset: 0x048
Reset: 0x00003FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			FML[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	FML[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 13:0 – FML[13:0] Frame Max Length
Rx jumbo frame maximum length.

62.8.20 GMAC System Bus Max Pipeline Register

Name: GMAC_AMP
Offset: 0x054
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								USE_FROM
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	AW2W_MAX_PIPELINE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AR2R_MAX_PIPELINE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – USE_FROM Address Write Bus to Write Data Bus Maximum Pipeline

Value	Name	Description
0	AW2W	Operates the AW2W_MAX_PIPELINE field between AW to W channel.
1	AW2B	Operates the AW2W_MAX_PIPELINE field between AW to B channel.

Bits 15:8 – AW2W_MAX_PIPELINE[7:0] Address Write Bus to Write Data Bus Maximum Pipeline
Defines the maximum number of outstanding write requests that can be issued by the DMA via the AW channel. This is effectively the write issuing capability.

Bits 7:0 – AR2R_MAX_PIPELINE[7:0] Address Read Bus to Read Data Bus Maximum Pipeline
Defines the maximum number of outstanding read requests that can be issued by the DMA via the AR channel. This is effectively the read issuing capability.

62.8.21 GMAC Interrupt Moderation Register

Name: GMAC_INTM
Offset: 0x05C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TXINTMOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXINTMOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – TXINTMOD[7:0] Transmit Interrupt Moderation
Count of 800 ns periods before bit 7 is set in GMAC_ISR.TCOMP. A non-zero value indicates transmit interrupt moderation will be performed.

Bits 7:0 – RXINTMOD[7:0] Receive Interrupt Moderation
Count of 800 ns periods before bit 1 is set in GMAC_ISR.RCOMP. A non-zero value indicates receive interrupt moderation will be performed.

62.8.22 GMAC System Wake-Up Time Register

Name: GMAC_SYSWT
Offset: 0x060
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SYSWKUPTIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SYSWKUPTIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SYSWKUPTIME[15:0] System Wake-Up Time
Count of 25.6 ns, 64 ns, 320 ns or 3200 ns intervals before transmission starts after deassertion of the bit RXLPISBC in Interrupt registers (each interval is equivalent to eight GTXCLK periods and varies with data rate).

62.8.23 GMAC Hash Register Bottom

Name: GMAC_HRB
Offset: 0x080
Reset: 0x00000000
Property: Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the Network Configuration Register (GMAC Network Configuration Register) enable the reception of hash matched frames. See Hash Addressing.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Hash Address
The first 32 bits of the Hash Address Register.

62.8.24 GMAC Hash Register Top

Name: GMAC_HRT
Offset: 0x084
Reset: 0x00000000
Property: Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the GMAC Network Configuration Register enable the reception of hash matched frames. See Hash Addressing.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Hash Address
Bits 63 to 32 of the Hash Address Register.

62.8.25 GMAC Specific Address 1 Bottom Register

Name: GMAC_SAB1
Offset: 0x088
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 1
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.26 GMAC Specific Address 1 Top Register

Name: GMAC_SAT1
Offset: 0x08C
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								FILTSORD
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – FILTSORD Filter Source or Destination MAC Address
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Description
0	The filter is a destination address filter.
1	The filter is a source address filter.

Bits 15:0 – ADDR[15:0] Specific Address 1
The most significant bits of the destination address, that is, bits 47:32.

62.8.27 GMAC Specific Address 2 Bottom Register

Name: GMAC_SAB2
Offset: 0x090
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 2
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.28 GMAC Specific Address 2 Top Register

Name: GMAC_SAT2
Offset: 0x094
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
			FILTBMASK[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								FILTSORD
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – FILTBMASK[5:0] Filter Bytes Mask
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Name	Description
`0x1`	BIT1	Controls whether the first byte has been received.
`0x2`	BIT2	Controls whether the second byte has been received.
`0x4`	BIT3	Controls whether the third byte has been received.
`0x8`	BIT4	Controls whether the fourth byte has been received.
`0x10`	BIT5	Controls whether the fifth byte has been received.
`0x20`	BIT6	Controls whether the sixth byte has been received

Bit 16 – FILTSORD Filter Source or Destination MAC Address
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Description
`0`	The filter is a destination address filter.
`1`	The filter is a source address filter.

Bits 15:0 – ADDR[15:0] Specific Address 2
The most significant bits of the destination address, that is, bits 47:32.

62.8.29 GMAC Specific Address 3 Bottom Register

Name: GMAC_SAB3
Offset: 0x098
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 3
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.30 GMAC Specific Address 3 Top Register

Name: GMAC_SAT3
Offset: 0x09C
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
			FILTBMASK[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								FILTSORD
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – FILTBMASK[5:0] Filter Bytes Mask
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Name	Description
0x1	BIT1	Controls whether the first byte has been received.
0x2	BIT2	Controls whether the second byte has been received.
0x4	BIT3	Controls whether the third byte has been received.
0x8	BIT4	Controls whether the fourth byte has been received.
0x10	BIT5	Controls whether the fifth byte has been received.
0x20	BIT6	Controls whether the sixth byte has been received

Bit 16 – FILTSORD Filter Source or Destination MAC Address
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Description
0	The filter is a destination address filter.
1	The filter is a source address filter.

Bits 15:0 – ADDR[15:0] Specific Address 3
The most significant bits of the destination address, that is, bits 47:32.

62.8.31 GMAC Specific Address 4 Bottom Register

Name: GMAC_SAB4
Offset: 0x0A0
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 4
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.32 GMAC Specific Address 4 Top Register

Name: GMAC_SAT4
Offset: 0x0A4
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
			FILTBMASK[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								FILTSORD
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – FILTBMASK[5:0] Filter Bytes Mask
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Name	Description
0x1	BIT1	Controls whether the first byte has been received.
0x2	BIT2	Controls whether the second byte has been received.
0x4	BIT3	Controls whether the third byte has been received.
0x8	BIT4	Controls whether the fourth byte has been received.
0x10	BIT5	Controls whether the fifth byte has been received.
0x20	BIT6	Controls whether the sixth byte has been received

Bit 16 – FILTSORD Filter Source or Destination MAC Address
Selects whether this filter should be comparing the MAC source address or the MAC destination address of the received Ethernet frame.

Value	Description
0	The filter is a destination address filter.
1	The filter is a source address filter.

Bits 15:0 – ADDR[15:0] Specific Address 4
The most significant bits of the destination address, that is, bits 47:32.

62.8.33 GMAC Type ID Match 1 Register

Name: GMAC_TIDM1
Offset: 0x0A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENID1							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENID1 Enable Copying of TID Matched Frames

Value	Description
0	TID is not part of the comparison match.
1	TID is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match 1
For use in comparisons with received frames type ID/length frames.

62.8.34 GMAC Type ID Match 2 Register

Name: GMAC_TIDM2
Offset: 0x0AC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENID2							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENID2 Enable Copying of TID Matched Frames

Value	Description
0	TID is not part of the comparison match.
1	TID is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match 2
For use in comparisons with received frames type ID/length frames.

62.8.35 GMAC Type ID Match 3 Register

Name: GMAC_TIDM3
Offset: 0x0B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENID3							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENID3 Enable Copying of TID Matched Frames

Value	Description
0	TID is not part of the comparison match.
1	TID is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match 3
For use in comparisons with received frames type ID/length frames.

62.8.36 GMAC Type ID Match 4 Register

Name: GMAC_TIDM4
Offset: 0x0B4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ENID4							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENID4 Enable Copying of TID Matched Frames

Value	Description
0	TID is not part of the comparison match.
1	TID is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match 4
For use in comparisons with received frames type ID/length frames.

62.8.37 GMAC Wake on LAN Register

Name: GMAC_WOL
Offset: 0x0B8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MTI	SA1	ARP	MAG
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 19 – MTI Multicast Hash Event Enable
Wake on LAN multicast hash event enable.

Bit 18 – SA1 Specific Address Register 1 Event Enable
Wake on LAN Specific Address Register 1 event enable.

Bit 17 – ARP ARP Request Event Enable
Wake on LAN ARP request event enable.

Bit 16 – MAG Magic Packet Event Enable
Wake on LAN magic packet event enable.

Bits 15:0 – IP[15:0] ARP Request IP Address
Wake on LAN ARP request IP address. Written to define the least significant 16 bits of the target IP address that is matched to generate a Wake on LAN event. A value of zero will not generate an event, even if this is matched by the received frame.

62.8.38 GMAC IPG Stretch Register

Name: GMAC_IPGS
Offset: 0x0BC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FL[15:0] Frame Length
Bits 7:0 are multiplied with the previously transmitted frame length (including preamble). Bits 15:8 +1 divide the frame length. If the resulting number is greater than 96 and bit 28 is set in the Network Configuration Register then the resulting number is used for the transmit inter-packet-gap. 1 is added to bits 15:8 to prevent a divide by zero. See MAC Transmit Block.

62.8.39 GMAC Stacked VLAN Register

Name: GMAC_SVLAN
Offset: 0x0C0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ESVLAN							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	VLAN_TYPE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VLAN_TYPE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ESVLAN Enable Stacked VLAN Processing Mode

Value	Description
0	Disable the stacked VLAN processing mode
1	Enable the stacked VLAN processing mode

Bits 15:0 – VLAN_TYPE[15:0] User Defined VLAN_TYPE Field
User defined VLAN_TYPE field. When Stacked VLAN is enabled, the first VLAN tag in a received frame will only be accepted if the VLAN type field is equal to this user defined VLAN_TYPE, OR equal to the standard VLAN type (0x8100). Note that the second VLAN tag of a Stacked VLAN packet will only be matched correctly if its VLAN_TYPE field equals 0x8100.

62.8.40 GMAC Transmit PFC Pause Register

Name: GMAC_TPFCP
Offset: 0x0C4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PQ[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PEV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PQ[7:0] Pause Quantum
If bit 17 of the Network Control Register is written with a one then for each entry equal to zero in the Transmit PFC Pause Register[15:8], the PFC pause frame's pause quantum field associated with that entry will be taken from the Transmit Pause Quantum Register. For each entry equal to one in the Transmit PFC Pause Register [15:8], the pause quantum associated with that entry will be zero.

Bits 7:0 – PEV[7:0] Priority Enable Vector
If bit 17 of the Network Control Register is written with a one then the priority enable vector of the PFC priority based pause frame will be set equal to the value stored in this register [7:0].

62.8.41 GMAC Specific Address 1 Mask Bottom Register

Name: GMAC_SAMB1
Offset: 0x0C8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 1 Mask
Setting a bit to one masks the corresponding bit in the Specific Address 1 Register.

62.8.42 GMAC Specific Address Mask 1 Top Register

Name: GMAC_SAMT1
Offset: 0x0CC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 1 Mask
Setting a bit to one masks the corresponding bit in the Specific Address 1 Register.

62.8.43 Address Mask for RX Data Buffer Accesses Register

Name: GMAC_AMRX
Offset: 0x0D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MSBADDR[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					MSBADDRMSK[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:28 – MSBADDR[3:0] MSB of the Receive Data Buffer Address
Values used to force bits 31:28 of the receive data buffer address to a particular value when the associated enable bits stored in this register [3:0] are set.
Any changes to this register are ignored while the DMA is processing a receive packet. It only affects the next full packet to be written to external system memory.

Bits 3:0 – MSBADDRMSK[3:0] Mask of the Receive Data Buffer Address
These bits are associated directly with bits[31:28].
When bit 0 is set, the address bit 28 used for accessing the receive data buffers will be forced to the value stored in bit 28 of this register.
When bit 1 is set, the address bit 29 used for accessing the receive data buffers will be forced to the value stored in bit 29 of this register.
When bit 2 is set, the address bit 30 used for accessing the receive data buffers will be forced to the value stored in bit 30 of this register.
When bit 3 is set, the address bit 31 used for accessing the receive data buffers will be forced to the value stored in bit 31 of this register.
When these bits are clear, the associated value stored in bits 31:28 have no effect on the address used for receive data buffer accesses.
Any changes to this registerare ignored while the DMA is processing a receive packet. It only affects the next full packet to be written to external memory.

62.8.44 PTP RX Unicast IP Destination Address Register

Name: GMAC_RXUDAR
Offset: 0x0D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RXUDA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXUDA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXUDA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXUDA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXUDA[31:0] Receive Unicast Destination Address
Unicast IP destination address used for detection of PTP frames on receive path.

62.8.45 PTP TX Unicast IP Destination Address Register

Name: GMAC_TXUDAR
Offset: 0x0D8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TXUDA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TXUDA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXUDA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXUDA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TXUDA[31:0] Transmit Unicast Destination Address
Unicast IP destination address used for detection of PTP frames on transmit path.

62.8.46 GMAC 1588 Timer Nanosecond Comparison Register

Name: GMAC_NSC
Offset: 0x0DC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			NANOSEC[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NANOSEC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NANOSEC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 21:0 – NANOSEC[21:0] 1588 Timer Nanosecond Comparison Value
Value is compared to the bits [45:24] of the TSU timer count value (upper 22 bits of nanosecond value).

62.8.47 GMAC 1588 Timer Second Comparison Low Register

Name: GMAC_SCL
Offset: 0x0E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SEC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SEC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SEC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SEC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEC[31:0] 1588 Timer Second Comparison Value
Value is compared to seconds value bits [31:0] of the TSU timer count value.

62.8.48 GMAC 1588 Timer Second Comparison High Register

Name: GMAC_SCH
Offset: 0x0E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SEC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SEC[15:0] 1588 Timer Second Comparison Value
Value is compared to the top 16 bits (most significant 16 bits [47:32] of seconds value) of the TSU timer count value.

62.8.49 GMAC PTP Event Frame Transmitted Seconds High Register

Name: GMAC_EFTSH
Offset: 0x0E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.50 GMAC PTP Event Frame Received Seconds High Register

Name: GMAC_EFRSH
Offset: 0x0EC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.51 GMAC PTP Peer Event Frame Transmitted Seconds High Register

Name: GMAC_PEFTSH
Offset: 0x0F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.52 GMAC PTP Peer Event Frame Received Seconds High Register

Name: GMAC_PEFRSH
Offset: 0x0F4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.53 GMAC Octets Transmitted Low Register

Name: GMAC_OTLO
Offset: 0x100
Reset: 0x00000000
Property: Read-only

When reading the Octets Transmitted and Octets Received Registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

Bit	31	30	29	28	27	26	25	24
	TXO[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TXO[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXO[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TXO[31:0] Transmitted Octets
Transmitted octets in frame without errors [31:0]. The number of octets transmitted in valid frames of any type. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.

62.8.54 GMAC Octets Transmitted High Register

Name: GMAC_OTHI
Offset: 0x104
Reset: 0x00000000
Property: Read-only

When reading the Octets Transmitted and Octets Received Registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TXO[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXO[15:0] Transmitted Octets
Transmitted octets in frame without errors [47:32]. The number of octets transmitted in valid frames of any type. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.

62.8.55 GMAC Frames Transmitted Register

Name: GMAC_FT
Offset: 0x108
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FTX[31:0] Frames Transmitted without Error
Frames transmitted without error. This register counts the number of frames successfully transmitted, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.56 GMAC Broadcast Frames Transmitted Register

Name: GMAC_BCFT
Offset: 0x10C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	BFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BFTX[31:0] Broadcast Frames Transmitted without Error
Broadcast frames transmitted without error. This register counts the number of broadcast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.57 GMAC Multicast Frames Transmitted Register

Name: GMAC_MFT
Offset: 0x110
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	MFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MFTX[31:0] Multicast Frames Transmitted without Error
This register counts the number of multicast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.58 GMAC Pause Frames Transmitted Register

Name: GMAC_PFT
Offset: 0x114
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PFTX[15:0] Pause Frames Transmitted Register
This register counts the number of pause frames transmitted. Only pause frames triggered by the register interface or through the external pause pins are counted as pause frames. Pause frames received through the FIFO interface are counted in the frames transmitted counter.

62.8.59 GMAC 64 Byte Frames Transmitted Register

Name: GMAC_BFT64
Offset: 0x118
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 64 Byte Frames Transmitted without Error
This register counts the number of 64 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.60 GMAC 65 to 127 Byte Frames Transmitted Register

Name: GMAC_TBFT127
Offset: 0x11C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 65 to 127 Byte Frames Transmitted without Error
This register counts the number of 65 to 127 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.61 GMAC 128 to 255 Byte Frames Transmitted Register

Name: GMAC_TBFT255
Offset: 0x120
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 128 to 255 Byte Frames Transmitted without Error
This register counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.62 GMAC 256 to 511 Byte Frames Transmitted Register

Name: GMAC_TBFT511
Offset: 0x124
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 256 to 511 Byte Frames Transmitted without Error
This register counts the number of 256 to 511 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.63 GMAC 512 to 1023 Byte Frames Transmitted Register

Name: GMAC_TBFT1023
Offset: 0x128
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 512 to 1023 Byte Frames Transmitted without Error
This register counts the number of 512 to 1023 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.64 GMAC 1024 to 1518 Byte Frames Transmitted Register

Name: GMAC_TBFT1518
Offset: 0x12C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 1024 to 1518 Byte Frames Transmitted without Error
This register counts the number of 1024 to 1518 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.65 GMAC Greater Than 1518 Byte Frames Transmitted Register

Name: GMAC_GTBFT1518
Offset: 0x130
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFTX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFTX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] Greater than 1518 Byte Frames Transmitted without Error
This register counts the number of 1518 or above byte frames successfully transmitted without error i.e., no underrun and not too many retries.

62.8.66 GMAC Transmit Underruns Register

Name: GMAC_TUR
Offset: 0x134
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXUNR[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	TXUNR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – TXUNR[9:0] Transmit Underruns
This register counts the number of frames not transmitted due to a transmit underrun. If this register is incremented then no other statistics register is incremented.

62.8.67 GMAC Single Collision Frames Register

Name: GMAC_SCF
Offset: 0x138
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SCOL[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCOL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – SCOL[17:0] Single Collision
This register counts the number of frames experiencing a single collision before being successfully transmitted i.e., no underrun.

62.8.68 GMAC Multiple Collision Frames Register

Name: GMAC_MCF
Offset: 0x13C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
							MCOL[17:16]	
Access							R	R
Reset							0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	MCOL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	MCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – MCOL[17:0] Multiple Collision
This register counts the number of frames experiencing between two and fifteen collisions prior to being successfully transmitted, i.e., no underrun and not too many retries.

62.8.69 GMAC Excessive Collisions Register

Name: GMAC_EC
Offset: 0x140
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							XCOL[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	XCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – XCOL[9:0] Excessive Collisions
This register counts the number of frames that failed to be transmitted because they experienced 16 collisions.

62.8.70 GMAC Late Collisions Register

Name: GMAC_LC
Offset: 0x144
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							LCOL[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – LCOL[9:0] Late Collisions
Counts the number of late collisions occurring after the slot time (512 bits) has expired. In 10/100 mode, late collisions are counted twice i.e., both as a collision and a late collision. In Gigabit mode, a late collision causes the transmission to be aborted, thus the single and multi collision registers are not updated.

62.8.71 GMAC Deferred Transmission Frames Register

Name: GMAC_DTF
Offset: 0x148
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DEFT[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DEFT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEFT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – DEFT[17:0] Deferred Transmission
This register counts the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.

62.8.72 GMAC Carrier Sense Errors Register

Name: GMAC_CSE
Offset: 0x14C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							CSR[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	CSR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – CSR[9:0] Carrier Sense Error
This register counts the number of frames transmitted where carrier sense was not seen during transmission or where carrier sense was deasserted after being asserted in a transmit frame without collision (no underrun). Only incremented in half duplex mode. The only effect of a carrier sense error is to increment this register. The behavior of the other statistics registers is unaffected by the detection of a carrier sense error.

62.8.73 GMAC Octets Received Low Register

Name: GMAC_ORLO
Offset: 0x150
Reset: 0x00000000
Property: Read-only

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

Bit	31	30	29	28	27	26	25	24
	RXO[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RXO[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXO[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXO[31:0] Received Octets
Received octets in frame without errors [31:0]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.74 GMAC Octets Received High Register

Name: GMAC_ORHI
Offset: 0x154
Reset: 0x00000000
Property: Read-only

When reading the Octets Transmitted and Octets Received Registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	RXO[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXO[15:0] Received Octets
Received octets in frame without errors [47:32]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.75 GMAC Frames Received Register

Name: GMAC_FR
Offset: 0x158
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FRX[31:0] Frames Received without Error
Frames received without error. This register counts the number of frames successfully received. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.76 GMAC Broadcast Frames Received Register

Name: GMAC_BCFR
Offset: 0x15C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	BFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BFRX[31:0] Broadcast Frames Received without Error
Broadcast frames received without error. This register counts the number of broadcast frames successfully received. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.77 GMAC Multicast Frames Received Register

Name: GMAC_MFR
Offset: 0x160
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	MFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MFRX[31:0] Multicast Frames Received without Error
This register counts the number of multicast frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.78 GMAC Pause Frames Received Register

Name: GMAC_PFR
Offset: 0x164
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PFRX[15:0] Pause Frames Received Register
This register counts the number of pause frames received without error.

62.8.79 GMAC 64 Byte Frames Received Register

Name: GMAC_BFR64
Offset: 0x168
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 64 Byte Frames Received without Error
This register counts the number of 64 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.80 GMAC 65 to 127 Byte Frames Received Register

Name: GMAC_TBFR127
Offset: 0x16C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 65 to 127 Byte Frames Received without Error
This register counts the number of 65 to 127 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.81 GMAC 128 to 255 Byte Frames Received Register

Name: GMAC_TBFR255
Offset: 0x170
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 128 to 255 Byte Frames Received without Error
This register counts the number of 128 to 255 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.82 GMAC 256 to 511 Byte Frames Received Register

Name: GMAC_TBFR511
Offset: 0x174
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 256 to 511 Byte Frames Received without Error
This register counts the number of 256 to 511 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.83 GMAC 512 to 1023 Byte Frames Received Register

Name: GMAC_TBFR1023
Offset: 0x178
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 512 to 1023 Byte Frames Received without Error
This register counts the number of 512 to 1023 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.84 GMAC 1024 to 1518 Byte Frames Received Register

Name: GMAC_TBFR1518
Offset: 0x17C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 1024 to 1518 Byte Frames Received without Error
This register counts the number of 1024 to 1518 byte frames successfully received without error, i.e., no underrun and not too many retries.

62.8.85 GMAC 1519 to Maximum Byte Frames Received Register

Name: GMAC_TMXBFR
Offset: 0x180
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	NFRX[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NFRX[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NFRX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 1519 to Maximum Byte Frames Received without Error
This register counts the number of 1519 byte or above frames successfully received without error. Maximum frame size is determined by the Network Configuration Register bit 8 (1536 maximum frame size) or bit 3 (jumbo frame size). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory. See GMAC Network Configuration Register.

62.8.86 GMAC Undersized Frames Received Register

Name: GMAC_UFR
Offset: 0x184
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							UFRX[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	UFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – UFRX[9:0] Undersize Frames Received
Counts the number of frames received less than 64 bytes in length (10/100 mode or Gigabit mode, full duplex) that do not have either a CRC error or an alignment error. In Gigabit mode, half duplex, this register counts either frames not conforming to the minimum slot time of 512 bytes or frames not conforming to the minimum frame size once bursting is active.

62.8.87 GMAC Oversized Frames Received Register

Name: GMAC_OFR
Offset: 0x188
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							OFRX[9:8]	
Access							R	R
Reset							0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	OFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – OFRX[9:0] Oversized Frames Received
This register counts the number of frames received exceeding 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register, 10240 bytes if bit 3 is set in the Network Configuration Register) in length but do not have either a CRC error, an alignment error nor a receive symbol error. See GMAC Network Configuration Register.

62.8.88 GMAC Jabbers Received Register

Name: GMAC_JR
Offset: 0x18C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							JRX[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	JRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – JRX[9:0] Jabbers Received
The register counts the number of frames received exceeding 1518 bytes in length (1536 if bit 8 is set in Network Configuration Register, 10240 bytes if bit 3 is set in the Network Configuration Register) and have either a CRC error, an alignment error or a receive symbol error. See GMAC Network Configuration Register.

62.8.89 GMAC Frame Check Sequence Errors Register

Name: GMAC_FCSE
Offset: 0x190
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							FCKR[9:8]	
Access							R	R
Reset							0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	FCKR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – FCKR[9:0] Frame Check Sequence Errors
The register counts frames that are an integral number of bytes, have bad CRC and are between 64 and 1518 bytes in length (1536 if bit 8 is set in Network Configuration Register, 10240 bytes if bit 3 is set in the Network Configuration Register). This register is also incremented if a symbol error is detected and the frame is of valid length and has an integral number of bytes.
This register is incremented for a frame with bad FCS, regardless of whether it is copied to memory due to ignore FCS mode being enabled in bit 26 of the Network Configuration Register. See GMAC Network Configuration Register.

62.8.90 GMAC Length Field Frame Errors Register

Name: GMAC_LFFE
Offset: 0x194
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							LFER[9:8]	
Access							R	R
Reset							0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LFER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – LFER[9:0] Length Field Frame Errors
This register counts the number of frames received that have a measured length shorter than that extracted from the length field (bytes 13 and 14). This condition is only counted if the value of the length field is less than 0x0600, the frame is not of excessive length and checking is enabled through bit 16 of the Network Configuration Register. See GMAC Network Configuration Register.

62.8.91 GMAC Receive Symbol Errors Register

Name: GMAC_RSE
Offset: 0x198
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXSE[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RXSE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – RXSE[9:0] Receive Symbol Errors
This register counts the number of frames that had GRXER asserted during reception. For 10/100 mode symbol errors are counted regardless of frame length checks. For Gigabit mode the frame must satisfy slot time requirements in order to count a symbol error. Additionally, in Gigabit half duplex mode, carrier extension errors are also recorded. Receive symbol errors will also be counted as an FCS or alignment error if the frame is between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register, 10240 bytes if bit 3 is set in the Network Configuration Register). If the frame is larger it will be recorded as a jabber error. See GMAC Network Configuration Register.

62.8.92 GMAC Alignment Errors Register

Name: GMAC_AE
Offset: 0x19C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							AER[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	AER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – AER[9:0] Alignment Errors
This register counts the frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of bytes and are between 64 and 1518 bytes in length (1536 if bit 8 is set in Network Configuration Register, 10240 bytes if bit 3 is set in the Network Configuration Register). This register is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of bytes. See GMAC Network Configuration Register.

62.8.93 GMAC Receive Resource Errors Register

Name: GMAC_RRE
Offset: 0x1A0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXERR[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	RXERR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXERR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – RXERR[17:0] Receive Resource Errors
Counts the frames that were successfully received by the MAC but could not be copied to memory because no receive buffer was available. This occurs when the GMAC reads a buffer descriptor with its ownership (or used) bit set.

62.8.94 GMAC Receive Overruns Register

Name: GMAC_ROE
Offset: 0x1A4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXOVR[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RXOVR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – RXOVR[9:0] Receive Overruns
This register counts the number of frames that are address recognized but were not copied to memory due to a receive overrun.

62.8.95 GMAC IP Header Checksum Errors Register

Name: GMAC_IHCE
Offset: 0x1A8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	HCKER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – HCKER[7:0] IP Header Checksum Errors
This register counts the number of frames discarded due to an incorrect IP header checksum, but are between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register or 10240 bytes if bit 3 is set in the Network Configuration Register) and do not have a CRC error, an alignment error, nor a symbol error.

62.8.96 GMAC TCP Checksum Errors Register

Name: GMAC_TCE
Offset: 0x1AC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TCKER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TCKER[7:0] TCP Checksum Errors
This register counts the number of frames discarded due to an incorrect TCP checksum, but are between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register or 10240 bytes if bit 3 is set in the Network Configuration Register) and do not have a CRC error, an alignment error, nor a symbol error.

62.8.97 GMAC UDP Checksum Errors Register

Name: GMAC_UCE
Offset: 0x1B0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	UCKER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – UCKER[7:0] UDP Checksum Errors
This register counts the number of frames discarded due to an incorrect UDP checksum, but are between 64 and 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register or 10240 bytes if bit 3 is set in the Network Configuration Register) and do not have a CRC error, an alignment error, nor a symbol error.

62.8.98 GMAC Flushed Received Packets Counter Register

Name: GMAC_FLRXPCR
Offset: 0x1B4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Flushed Received Packets Count (cleared on read)
Counts the number of frames that have been flushed from the receive packet buffer memory due to one of the following reasons:

- When partial store and forward mode is enabled and a packet is received while there is no system bus resource
- When partial store and forward mode is enabled and a system bus error is encountered while writing the packet data to system memory.
- When automatic discard of received packed during lack of resource is enabled (bit 24 of the DMA Configuration register) and a packet is received while there is no system bus resource.
- When a software flush of a packet from the head of the packet buffer queue (bit 18 of the Network Control register) is performed and the DMA is not currently busy.”

62.8.99 GMAC 1588 Timer Increment Sub-nanoseconds Register

Name: GMAC_TISUBN
Offset: 0x1BC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	LSBTIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MSBTIR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBTIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – LSBTIR[7:0] Lower Significant Bits of Timer Increment Register
Lower significant bits of Timer Increment Register[15:0] giving a 24-bit timer_increment counter. These bits are the sub-ns value which the 1588 timer will be incremented each clock cycle. Bit n = $2^{(n-16)}$ nsec giving a resolution of approximately $15.2E^{-15}$ sec.

Bits 31:24 – LSBTIR[7:0] Lower Significant Bits of Timer Increment Register
Lower significant bits of Timer Increment Register[15:0] giving a 24-bit timer_increment counter. These bits are the sub-ns value which the 1588 timer will be incremented each clock cycle. Bit n = $2^{(n-16)}$ nsec giving a resolution of approximately $15.2E^{-15}$ sec.

Bits 15:0 – MSBTIR[15:0] Most Significant Bits of Timer Increment Register
Most significant bits [23:8] of the sub-nanosecond value by which the 1588 timer will be incremented each clock cycle. 24 bits of sub-nanosecond precision gives a resolution of approximately $5.86E^{-17}$ seconds.

62.8.100 GMAC 1588 Timer Seconds High Register

Name: GMAC_TSH
Offset: 0x1C0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TCS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TCS[15:0] Timer Count in Seconds
This register is writable. It increments by one when the 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust register is written.

62.8.101 GMAC 1588 Timer Seconds Low Register

Name: GMAC_TSL
Offset: 0x1D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TCS[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TCS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TCS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TCS[31:0] Timer Count in Seconds
This register is writable. It increments by one when the 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.

62.8.102 GMAC 1588 Timer Nanoseconds Register

Name: GMAC_TN
Offset: 0x1D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			TNS[29:24]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TNS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TNS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TNS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – TNS[29:0] Timer Count in Nanoseconds
This register is writable. It can also be adjusted by writes to the 1588 Timer Adjust Register. It increments by the value of the 1588 Timer Increment Register each clock cycle.

62.8.103 GMAC 1588 Timer Adjust Register

Name: GMAC_TA
Offset: 0x1D8
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	ADJ		ITDT[29:24]					
Access	W		W	W	W	W	W	W
Reset	–		–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	ITDT[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	ITDT[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ITDT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – ADJ Adjust 1588 Timer
Write as one to subtract from the 1588 timer. Write as zero to add to it.

Bits 29:0 – ITDT[29:0] Increment/Decrement
The number of nanoseconds to increment or decrement the 1588 Timer Nanoseconds Register. If necessary, the 1588 Seconds Register will be incremented or decremented.

62.8.104 GMAC 1588 Timer Increment Register

Name: GMAC_TI
Offset: 0x1DC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	NIT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ACNS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CNS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – NIT[7:0] Number of Increments
The number of increments after which the alternative increment is used.

Bits 15:8 – ACNS[7:0] Alternative Count Nanoseconds
Alternative count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

Bits 7:0 – CNS[7:0] Count Nanoseconds
A count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

62.8.105 GMAC PTP Event Frame Transmitted Seconds Low Register

Name: GMAC_EFTSL
Offset: 0x1E0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the 1588 Timer Seconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.106 GMAC PTP Event Frame Transmitted Nanoseconds Register

Name: GMAC_EFTN
Offset: 0x1E4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.107 GMAC PTP Event Frame Received Seconds Low Register

Name: GMAC_EFRSL
Offset: 0x1E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.108 GMAC PTP Event Frame Received Nanoseconds Register

Name: GMAC_EFRN
Offset: 0x1EC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.109 GMAC PTP Peer Event Frame Transmitted Seconds Low Register

Name: GMAC_PEFTSL
Offset: 0x1F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the 1588 Timer Seconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.110 GMAC PTP Peer Event Frame Transmitted Nanoseconds Register

Name: GMAC_PEFTN
Offset: 0x1F4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.111 GMAC PTP Peer Event Frame Received Seconds Low Register

Name: GMAC_PEFRSL
Offset: 0x1F8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RUD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.112 GMAC PTP Peer Event Frame Received Nanoseconds Register

Name: GMAC_PEFRN
Offset: 0x1FC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RUD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

62.8.113 GMAC Transmit Pause Quantum 1 Register

Name: GMAC_TXPQUANT1
Offset: 0x260
Reset: 0xFFFFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	QUANT_P3[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	QUANT_P3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	QUANT_P2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	QUANT_P2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:16 – QUANT_P3[15:0] Priority 3 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 3.

Bits 15:0 – QUANT_P2[15:0] Priority 2 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 2.

62.8.114 GMAC Transmit Pause Quantum 2 Register

Name: GMAC_TXPQUANT2
Offset: 0x264
Reset: 0xFFFFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	QUANT_P5[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	QUANT_P5[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	QUANT_P4[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	QUANT_P4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:16 – QUANT_P5[15:0] Priority 5 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 5.

Bits 15:0 – QUANT_P4[15:0] Priority 4 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 4.

62.8.115 GMAC Transmit Pause Quantum 3 Register

Name: GMAC_TXPQUANT3
Offset: 0x268
Reset: 0xFFFFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	QUANT_P7[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	QUANT_P7[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	QUANT_P6[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	QUANT_P6[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:16 – QUANT_P7[15:0] Priority 7 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 7.

Bits 15:0 – QUANT_P6[15:0] Priority 6 Transmit Pause Quantum
Transmit pause quantum written with the pause quantum value for pause frame transmission of priority 6.

62.8.116 GMAC Received LPI Transitions

Name: GMAC_RXLPI
Offset: 0x270
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Count of Received LPI Transitions (cleared on read)
A count of the number of times there is a transition from receiving normal idle to receiving low power idle.

62.8.117 GMAC Received LPI Time

Name: GMAC_RXLPITIME
Offset: 0x274
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	LPITIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	LPITIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LPITIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – LPITIME[23:0] Time in LPI (cleared on read)
This field increments once every 16 MCK cycles when the bit LPI Indication (bit 7) is set in the Network Status register.

62.8.118 GMAC Transmit LPI Transitions

Name: GMAC_TXLPI
Offset: 0x278
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Count of LPI transitions (cleared on read)
A count of the number of times the bit Enable LPI Transmission (bit 19) goes from low to high in the Network Control register.

62.8.119 GMAC Transmit LPI Time

Name: GMAC_TXLPITIME
Offset: 0x27C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	LPITIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	LPITIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	LPITIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – LPITIME[23:0] Time in LPI (cleared on read)
This field increments once every 16 MCK cycles when the bit Enable LPI Transmission (bit 19) is set in the Network Control register.

62.8.120 GMAC Quality of Service Configuration Register 0

Name: GMAC_QOS_CFG0
Offset: 0x2E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	Q3_DESCR[3:0]				Q3_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Q2_DESCR[3:0]				Q2_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Q1_DESCR[3:0]				Q1_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Q0_DESCR[3:0]				Q0_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – Q3_DESCR[3:0] System Bus QoS Attributes for Queue 3 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 3.

Bits 27:24 – Q3_DATA[3:0] System Bus QoS Attributes for Queue 3 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 2.

Bits 23:20 – Q2_DESCR[3:0] System Bus QoS Attributes for Queue 2 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 2.

Bits 19:16 – Q2_DATA[3:0] System Bus QoS Attributes for Queue 2 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 2.

Bits 15:12 – Q1_DESCR[3:0] System Bus QoS Attributes for Queue 1 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 1.

Bits 11:8 – Q1_DATA[3:0] System Bus QoS Attributes for Queue 1 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 1.

Bits 7:4 – Q0_DESCR[3:0] System Bus QoS Attributes for Queue 0 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 0.

Bits 3:0 – Q0_DATA[3:0] System Bus QoS Attributes for Queue 0 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 0.

62.8.121 GMAC Quality of Service Configuration Register 1

Name: GMAC_QOS_CFG1
Offset: 0x2E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	Q6_DESCR[3:0]				Q6_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Q5_DESCR[3:0]				Q5_DATA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – Q6_DESCR[3:0] System Bus QoS Attributes for Queue 6 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 6.

Bits 11:8 – Q6_DATA[3:0] System Bus QoS Attributes for Queue 6 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 6.

Bits 7:4 – Q5_DESCR[3:0] System Bus QoS Attributes for Queue 5 Descriptor Access
Defines the value passed on the system bus QoS attributes when accessing the descriptor of queue 5.

Bits 3:0 – Q5_DATA[3:0] System Bus QoS Attributes for Queue 5 Data Access
Defines the value passed on the system bus QoS attributes when accessing the data of queue 5.

62.8.122 GMAC Interrupt Status Register Priority Queue x

Name: GMAC_ISRPQx
Offset: 0x0400 + (x-1)*0x04 [x=1..5]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HRESP			
Access					R/W			
Reset					0			

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX			RXUBR	RCOMP	
Access	R/W	R/W	R/W			R/W	R/W	
Reset	0	0	0			0	0	

Bit 11 – HRESP System Bus Error

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error
Set if an error occurs whilst midway through reading transmit frame from the system bus, including system buss errors and buffers exhausted mid frame.

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

62.8.123 GMAC Transmit Buffer Queue Base Address Register Priority Queue x

Name: GMAC_TBQBAPQx
Offset: 0x0440 + (x-1)*0x04 [x=1..5]
Reset: 0x00000000
Property: Read/Write

These registers hold the start address of the transmit buffer queues (transmit buffers descriptor lists) for the additional queues and must be initialized to the address of valid descriptors, even if the priority queues are not used.

Bit	31	30	29	28	27	26	25	24
	TXBQBA[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TXBQBA[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TXBQBA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TXBQBA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – TXBQBA[29:0] Transmit Buffer Queue Base Address
Written with the address of the start of the transmit queue.

62.8.124 GMAC Receive Buffer Queue Base Address Register Priority Queue x

Name: GMAC_RBQBAPQx
Offset: 0x0480 + (x-1)*0x04 [x=1..5]
Reset: 0x00000000
Property: Read/Write

These registers hold the start address of the receive buffer queues (receive buffers descriptor lists) for the additional queues and must be initialized to the address of valid descriptors, even if the priority queues are not used.

Bit	31	30	29	28	27	26	25	24
	RXBQBA[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXBQBA[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXBQBA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXBQBA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – RXBQBA[29:0] Receive Buffer Queue Base Address
Written with the address of the start of the receive queue.

62.8.125 GMAC Receive Buffer Size Register Priority Queue x

Name: GMAC_RBSRPQx
Offset: 0x04A0 + (x-1)*0x04 [x=1..5]
Reset: 0x00000002
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bits 7:0 – RBS[7:0] Receive Buffer Size
DMA receive buffer size in system memory. The value defined by these bits determines the size of buffer to use in main system memory when writing received data.
The value is defined in multiples of 64 bytes such that a value of 0x01 corresponds to buffers of 64 bytes, 0x02 corresponds to 128 bytes etc.
For example:
0x02: 128 bytes
0x18: 1536 bytes (1 × max length frame/buffer)
0xA0: 10240 bytes (1 × 10K jumbo frame/buffer)
Note that this value should never be written as zero.

62.8.126 GMAC Credit-Based Shaping Control Register

Name: GMAC_CBSCR
Offset: 0x4BC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							QBE	QAE
Access							R/W	R/W
Reset							0	0

Bit 1 – QBE Queue B CBS Enable

Value	Description
0	Credit-based shaping on the highest priority queue (queue B) is disabled.
1	Credit-based shaping on the highest priority queue (queue B) is enabled.

Bit 0 – QAE Queue A CBS Enable

Value	Description
0	Credit-based shaping on the second highest priority queue (queue A) is disabled.
1	Credit-based shaping on the second highest priority queue (queue A) is enabled.

62.8.127 GMAC Credit-Based Shaping IdleSlope Register for Queue A

Name: GMAC_CBSISQA
Offset: 0x4C0
Reset: 0x00000000
Property: Read/Write

Credit-based shaping must be disabled in GMAC_CBSCR before updating this register.

Bit	31	30	29	28	27	26	25	24
	IS[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IS[31:0] IdleSlope
IdleSlope value for queue A in bytes/second.
The IdleSlope value is defined as the rate of change of credit when a packet is waiting to be sent. This must not exceed the port transmit rate which is dependent on the speed of operation, e.g., 100 Mb/second = 0x017D7840
If 50% of bandwidth was to be allocated to a particular queue in 100 Mb/second mode, then the IdleSlope value for that queue would be calculated as 0x017D7840/2.

62.8.128 GMAC Credit-Based Shaping IdleSlope Register for Queue B

Name: GMAC_CBSISQB
Offset: 0x4C4
Reset: 0x00000000
Property: Read/Write

Credit-based shaping must be disabled in GMAC_CBSCR before updating this register.

Bit	31	30	29	28	27	26	25	24
	IS[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IS[31:0] IdleSlope
IdleSlope value for queue B in bytes/second.
The IdleSlope value is defined as the rate of change of credit when a packet is waiting to be sent. This must not exceed the port transmit rate which is dependent on the speed of operation, e.g., 100 Mb/second = 0x017D7840.
If 50% of bandwidth was to be allocated to a particular queue in 100 Mb/second mode, then the IdleSlope value for that queue would be calculated as 0x017D7840/2.

62.8.129 GMAC Transmit Queue Upper Buffer Address Register

Name: GMAC_TQUBA
Offset: 0x4C8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TQUBA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TQUBA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TQUBA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TQUBA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TQUBA[31:0] Transmit Queue Upper Buffer Address
Upper 32 bits of transmit buffer descriptor queue base address.

62.8.130 GMAC Transmit Buffer Data Control Register

Name: GMAC_TXBDCTRL
Offset: 0x4CC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TSMODE[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bits 5:4 – TSMODE[1:0] Transmit Descriptor Timestamp Insertion Mode

Value	Name	Description
0	DISABLE	Timestamp insertion disable.
1	PTPEVENT	Timestamp inserted for PTP Event Frames only.
2	PTPALL	Timestamp inserted for All PTP Frames only.
3	ALL	Timestamp inserted for All Frames.

62.8.131 GMAC Receive Buffer Data Control Register

Name: GMAC_RXBDCTRL
Offset: 0x4D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TSMODE[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bits 5:4 – TSMODE[1:0] Receive Descriptor Timestamp Insertion Mode

Value	Name	Description
0	DISABLE	Timestamp insertion disable.
1	PTPEVENT	Timestamp inserted for PTP Event Frames only.
2	PTPALL	Timestamp inserted for All PTP Frames only.
3	ALL	Timestamp inserted for All Frames.

62.8.132 GMAC Receive Queue Upper Buffer Address Register

Name: GMAC_RQUBA
Offset: 0x4D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RQUBA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RQUBA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RQUBA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RQUBA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RQUBA[31:0] Receive Queue Upper Buffer Address
Upper 32 bits of receive buffer descriptor queue base address.

62.8.133 GMAC Screening Type 1 Register x Priority Queue

Name: GMAC_ST1RPQx
Offset: 0x0500 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Screening type 1 registers are used to allocate up to 6 priority queues to received frames based on certain IP or UDP fields of incoming frames.

Bit	31	30	29	28	27	26	25	24
			UDPE	DSTCE	UDPM[15:12]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UDPM[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UDPM[3:0]				DSTCM[7:4]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DSTCM[3:0]					QNB[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 29 – UDPE UDP Port Match Enable
When UDP port match enable is set (bit 29), the UDP Destination Port of the received UDP frame is matched against bits 27:12.

Bit 28 – DSTCE Differentiated Services or Traffic Class Match Enable
When DS/TC match enable is set (bit 28), the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against bits 11:4.

Bits 27:12 – UDPM[15:0] UDP Port Match
When UDP port match enable is set (bit 29), the UDP Destination Port of the received UDP frame is matched against bits 27:12.

Bits 11:4 – DSTCM[7:0] Differentiated Services or Traffic Class Match
When DS/TC match enable is set (bit 28), the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against bits 11:4.

Bits 2:0 – QNB[2:0] Queue Number (0–5)
If a match is successful, then the queue value programmed in bits 2:0 is allocated to the frame.

62.8.134 GMAC Screening Type 2 Register x Priority Queue

Name: GMAC_ST2RPQx
Offset: 0x0540 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Screening type 2 registers are used to allocate up to 6 priority queues to received frames based on the VLAN priority field of received Ethernet frames.

Bit	31	30	29	28	27	26	25	24
		COMPCE	COMPC[4:0]					COMPBE
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMPB[4:0]					COMPAE	COMPA[4:3]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		VLANP[2:0]				QNB[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 30 – COMPCE Compare C Enable

Value	Description
0	Comparison via the register designated by index COMPC is disabled.
1	Comparison via the register designated by index COMPC is enabled.

Bits 29:25 – COMPC[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPC is a pointer to the compare registers GMAC_ST2CW0Rx and GMAC_ST2CW1Rx. When COMPCE is set, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 24 – COMPBE Compare B Enable

Value	Description
0	Comparison via the register designated by index COMPB is disabled.
1	Comparison via the register designated by index COMPB is enabled.

Bits 23:19 – COMPB[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPB is a pointer to the compare registers GMAC_ST2CW0Rx and GMAC_ST2CW1Rx. When COMPBE is set, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 18 – COMPAE Compare A Enable

Value	Description
0	Comparison via the register designated by index COMPA is disabled.
1	Comparison via the register designated by index COMPA is enabled.

Bits 17:13 – COMPA[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPA is a pointer to the compare registers GMAC_ST2CW0Rx and GMAC_ST2CW1Rx. When COMPAE is set, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 12 – ETHE EtherType Enable

Value	Description
0	EtherType match with bits 15:0 in the register designated by the value of I2ETH is disabled.
1	EtherType match with bits 15:0 in the register designated by the value of I2ETH is enabled.

Bits 11:9 – I2ETH[2:0] Index of Screening Type 2 EtherType register x
When ETHE is set (bit 12), the field EtherType (last EtherType in the header if the frame is VLAN tagged) is compared with bits 15:0 in the register designated by the value of I2ETH.

Bit 8 – VLANE VLAN Enable

Value	Description
0	VLAN match is disabled.
1	VLAN match is enabled.

Bits 6:4 – VLANP[2:0] VLAN Priority
When VLAN match enable is set (bit 8), the VLAN priority field of the received frame is matched against bits 7:4 of this register.

Bits 2:0 – QNB[2:0] Queue Number (0–5)
If a match is successful, then the queue value programmed in QNB is allocated to the frame.

62.8.135 GMAC Transmit Schedule Control Register

Name: GMAC_TSCTL
Offset: 0x580
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					TXSQ5[1:0]		TXSQ4[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXSQ3[1:0]		TXSQ2[1:0]		TXSQ1[1:0]		TXSQ0[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:1, 2:3, 4:5, 6:7, 8:9, 10:11 – TXSQx Transmit Schedule for Qx

Value	Description
0	Fixed priority.
1	CBS Enabled only valid for top two enabled queues and if CBS capability selected.
2	DWRR enabled
3	ETS enabled

62.8.136 GMAC Transmit Queue Bandwidth Rate Limit 0 Register

Name: GMAC_TQBWRL0
Offset: 0x590
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ALLOCQ3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ALLOCQ2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ALLOCQ1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ALLOCQ0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – ALLOCQx DWRR Weighting or ETS Bandwidth Allocation for Qx
Defines the value of Deficit Weighted Round Robin (DWRR) or Enhanced Transmission Selection (ETS - 802.1Qaz).

62.8.137 GMAC Transmit Queue Bandwidth Rate Limit 1 Register

Name: GMAC_TQBWRL1
Offset: 0x594
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ALLOCQ5[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ALLOCQ4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15 – ALLOCQx DWRR Weighting or ETS Bandwidth Allocation for Qx
Defines the value of Deficit Weighted Round Robin (DWRR) or Enhanced Transmission Selection (ETS - 802.1Qaz).

62.8.138 GMAC Transmit Queue Segment Allocation Register

Name: GMAC_TQSA
Offset: 0x5A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SEGALLOCQ5[2:0]				SEGALLOCQ4[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
		SEGALLOCQ3[2:0]				SEGALLOCQ2[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		SEGALLOCQ1[2:0]				SEGALLOCQ0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22 – SEGALLOCQx Segment Allocation for Qx
Number of segments allocated to Qx. This should be entered as a log 2; for example, entering a value of 2 grants 4 segments. A maximum of 32 segments can be granted.

62.8.139 GMAC Interrupt Enable Register Priority Queue x

Name: GMAC_IERPQx
Offset: 0x0600 + (x-1)*0x04 [x=1..5]
Reset: –
Property: Write-only

The following values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HRESP			
Access					W			
Reset					–			

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX			RXUBR	RCOMP	
Access	W	W	W			W	W	
Reset	–	–	–			–	–	

Bit 11 – HRESP System Bus Error

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

62.8.140 GMAC Interrupt Disable Register Priority Queue x

Name: GMAC_IDRPQx
Offset: 0x0620 + (x-1)*0x04 [x=1..5]
Reset: –
Property: Write-only

The following values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HRESP			
Access					W			
Reset					–			

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX			RXUBR	RCOMP	
Access	W	W	W			W	W	
Reset	–	–	–			–	–	

Bit 11 – HRESP System Bus Error

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

62.8.141 GMAC Interrupt Mask Register Priority Queue x

Name: GMAC_IMRPQx
Offset: 0x0640 + (x-1)*0x04 [x=1..5]
Reset: 0x00000000
Property: Read/Write

A read of this register returns the value of the receive complete interrupt mask.

A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

The following values are valid for all listed bit names of this register:

0: Corresponding interrupt is enabled.

1: Corresponding interrupt is disabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HRESP			
Access					R/W			
Reset					0			

Bit	7	6	5	4	3	2	1	0
	TCOMP	AHB	RLEX			RXUBR	RCOMP	
Access	R/W	R/W	R/W			R/W	R/W	
Reset	0	0	0			0	0	

Bit 11 – HRESP System Bus Error

Bit 7 – TCOMP Transmit Complete

Bit 6 – AHB Transmit Frame Corruption Due to System Bus Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

62.8.142 GMAC Screening Type 2 EtherType Register x

Name: GMAC_ST2ERx
Offset: 0x06E0 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COMPVAL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMPVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMPVAL[15:0] Ethertype Compare Value
When the bit GMAC_ST2RPQ.ETHE is enabled, the EtherType (last EtherType in the header if the frame is VLAN tagged) is compared with bits 15:0 in the register designated by GMAC_ST2RPQ.I2ETH.

62.8.143 GMAC Screening Type 2 Compare Word 0 Register x

Name: GMAC_ST2CW0Rx
Offset: 0x0700 + x*0x08 [x=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	COMPVAL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMPVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MASKVAL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MASKVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – COMPVAL[15:0] Compare Value
The byte stored in bits [23:16] is compared against the first byte of the 2 bytes extracted from the frame.
The byte stored in bits [31:24] is compared against the second byte of the 2 bytes extracted from the frame.

Bits 15:0 – MASKVAL[15:0] Mask Value
The value of MASKVAL ANDed with the 2 bytes extracted from the frame is compared to the value of MASKVAL ANDed with the value of COMPVAL.

62.8.144 GMAC Screening Type 2 Compare Word 1 Register x

Name: GMAC_ST2CW1Rx
Offset: 0x0704 + x*0x08 [x=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DISMASK	OFFSSTRT[1]
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	OFFSSTRT[0]	OFFSVAL[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 9 – DISMASK Disable Mask
Controls whether GMAC_ST2CW0R contains a 2-byte compare value with a 2-byte mask value or a 4-byte compare value.

Value	Description
0	GMAC_ST2CW0Rx contains a 2-byte compare value with a 2-byte mask value.
1	GMAC_ST2CW0Rx contains a 4-byte compare value.

Bits 8:7 – OFFSSTRT[1:0] Ethernet Frame Offset Start

Value	Name	Description
0	FRAMESTART	Offset from the start of the frame
1	ETHERTYPE	Offset from the byte after the EtherType field
2	IP	Offset from the byte after the IP header field
3	TCP_UDP	Offset from the byte after the TCP/UDP header field

Bits 6:0 – OFFSVAL[6:0] Offset Value in Bytes
The value of OFFSVAL ranges from 0 to 127 bytes, and is counted from either the start of the frame, the byte after the EtherType field (last EtherType in the header if the frame is VLAN tagged), the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header.

63. Flexible Serial Communication Controller (FLEXCOM)

63.1 Description

The Flexible Serial Communication Controller (FLEXCOM) offers several serial communication protocols that are managed by the three submodules USART, SPI, and TWI (I2C).

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full-duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver timeout enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART features three test modes: Remote Loopback, Local Loopback and Automatic Echo.

The USART supports specific operating modes providing interfaces on RS485, LIN, , with ISO7816 T = 0 or T = 1 smart card slots, and infrared transceivers. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS.

The USART supports the connection to the DMA Controller, which enables data transfers to the transmitter and from the receiver. The DMAC provides chained buffer management without any intervention of the processor.

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Host or Client mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “host” which controls the data flow, while the other devices act as “clients” which have data shifted into and out by the host. Different CPUs can take turn being hosts (multiple host protocol, contrary to single host protocol where one CPU is always the host while all of the others are always clients). One host can simultaneously shift data into multiple clients. However, only one client can drive its output to write data back to the host at any given time.

A client device is selected when the host asserts its NSS signal. If multiple client devices exist, the host generates a separate client select signal for each client (NPCS).

The SPI system consists of two data lines and two control lines:

- Host Out Client In (MOSI)—This data line supplies the output data from the host shifted into the input(s) of the client(s).
- Host In Client Out (MISO)—This data line supplies the output data from a client to the input of the host. There may be no more than one client transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the host and regulates the flow of the data bits. The host can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Client Select (NSS)—This control line allows clients to be turned on and off by hardware.

The Two-wire Interface (TWI) interconnects components on a unique two-wire bus, made up of one clock line and one data line based on a byte-oriented transfer format. It can be used with any Two-wire Interface bus Serial EEPROM and I2C-compatible devices, such as a Real-Time Clock (RTC), Dot Matrix/Graphic LCD Controller and temperature sensor. The TWI is programmable as a host or a client with sequential or single-byte access. Multiple host capability is supported.

Arbitration of the bus is performed internally and puts the TWI in Client mode automatically if the bus arbitration is lost.

A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

The following table lists the compatibility level of any TWI in Host mode and a full I2C compatible device.

Table 63-1. TWI Compatibility with I2C Standard

I2C Standard	TWI
Standard mode speed (100 kHz)	Host, Multi-Host, Client supported
Fast mode speed (400 kHz)	Host, Multi-Host, Client supported
Fast mode Plus speed (1 MHz)	Host, Multi-Host, Client supported
High-speed mode (3.4 MHz)	Host, Client supported
7- or 10-bit[(1)] Client addressing	Supported
Repeated Start (Sr) condition	Supported
ACK and NACK management	Supported
Input filtering	Supported
Slope control	Not supported
Clock stretching	Supported

Note:
1. 10-bit support in Host mode only.

63.2 Embedded Characteristics

63.2.1 USART/UART Characteristics

- 32-data Transmit and Receive FIFOs
- Programmable Baud Rate Generator
- Baud Rate can be Independent of the Processor/Peripheral Clock
- Supports Asynchronous Partial Wakeup on Receive Line Activity
- Comparison Function on Received Character
- 5-bit to 9-bit Full-duplex Synchronous or Asynchronous Serial Communications
  - 1, 1.5 or 2 stop bits in Asynchronous mode or 1 or 2 stop bits in Synchronous mode
  - Parity generation and error detection
  - Framing error detection, overrun error detection
  - Digital filter on receive line
  - MSB- or LSB-first
  - Optional break generation and detection
  - By 8 or by 16 oversampling receiver frequency
  - Optional hardware handshaking RTS-CTS
  - Receiver timeout and transmitter timeguard
  - Optional Multidrop mode with address generation and detection
- RS485 with Driver Control Signal
- ISO7816, T = 0 or T = 1 Protocols for Interfacing with Smart Cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA Modulation and Demodulation
  - Communication at up to 115.2 kbit/s
- Up to 16-bit data, Manchester-encoded Frame Support
- LIN Mode
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Host or client
  - Processing of frames with up to 256 data bytes

  - Response data length can be configurable or defined automatically by the identifier
  - Self-synchronization in client node configuration
  - Automatic processing and verification of the “synch break” and the “synch field”
  - “Synch break” detection even when partially superimposed with a data byte
  - Automatic identifier parity calculation/sending and verification
  - Parity sending and verification can be disabled
  - Automatic checksum calculation/sending and verification
  - Checksum sending and verification can be disabled
  - Support both “classic” and “enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot mode: host allocates slots to the scheduled frames automatically
  - Generation of the wakeup signal
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- Supports Connection of:
  - Two DMA Controller (DMAC) channels
  - Offers buffer transfer without processor intervention
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access

63.2.2 SPI Characteristics

- 32-data Transmit and Receive FIFOs
- Host or Client Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
- Selectable Mode Fault Detection
- Host Mode Can Drive SPCK up to Peripheral Clock
- Host Mode Bit Rate Can Be Independent of the Processor/Peripheral Clock
- Client Mode Operates on SPCK, Asynchronously with Core and Bus Clock
- Four Chip Selects with External Decoder Support Allow Communication with up to 15Peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to DMA Channels Optimizes Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access

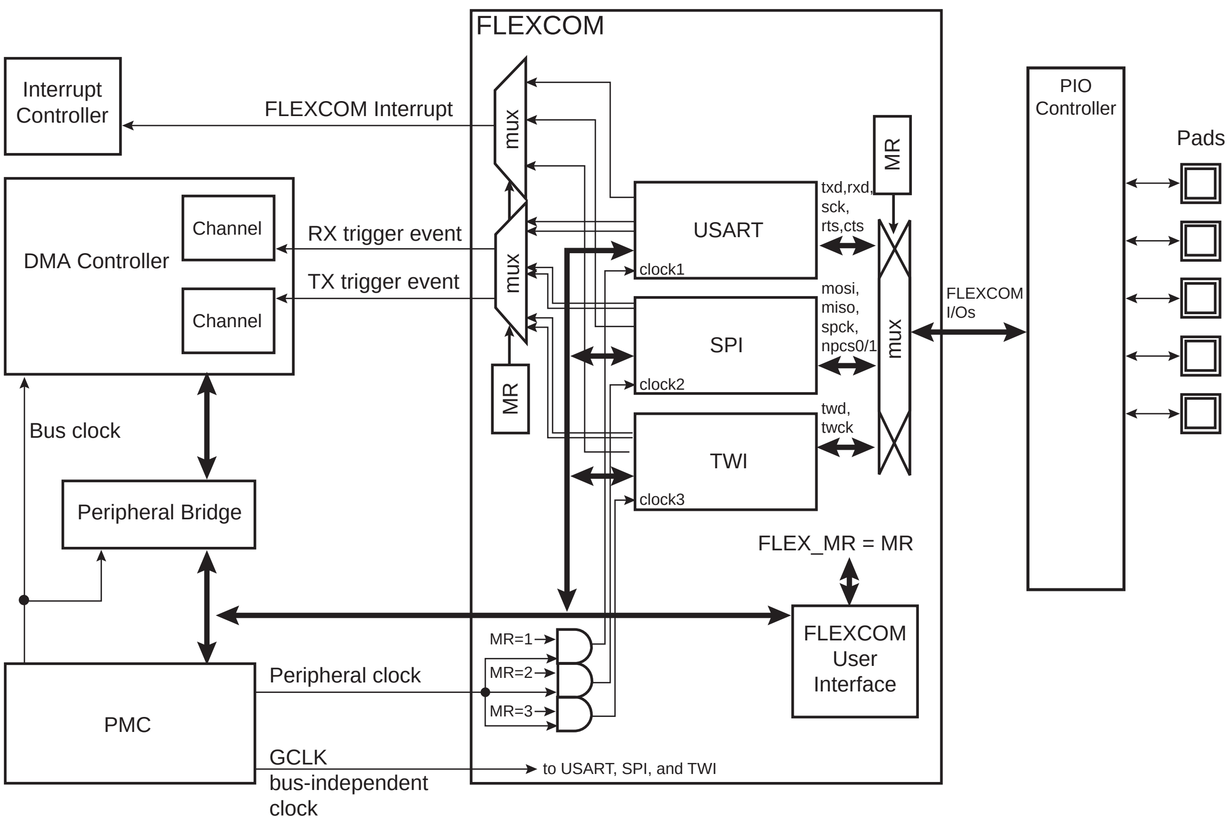
63.2.3 TWI/SMBus Characteristics

- 32-byte Transmit and Receive FIFOs
- Bit Rate can be Independent of the Processor/Peripheral Clock
- SMBus Support
- Compatible with I^2C Compatible Devices[(1)]
- One, Two or Three Bytes for Client Address

- Sequential Read/Write Operations
- General Call Supported in Client Mode
- Connection to DMA Controller Channels Optimizes Data Transfers
  – One channel for the receiver
  – One channel for the transmitter
- Functional Safety: Protection, Monitors and Reports
  – Register Write protection
  – Reports any write-protected access
- **Note:**
  1. See table TWI Compatibility with I2C Standard for further details.

63.3 Block Diagram

Figure 63-1. FLEXCOM Block Diagram



63.4 I/O Lines Description

Table 63-2. I/O Lines Description

Name	Description			Type
	USART/UART	SPI	TWI	
FLEXCOM_IO0	TXD	MOSI	TWD	I/O
FLEXCOM_IO1	RXD	MISO	TWCK	I/O
FLEXCOM_IO2	SCK	SPCK	–	I/O

...........continued

Name	Description			Type
	USART/UART	SPI	TWI	
FLEXCOM_IO3	CTS	NPCS0/NSS	–	I/O
FLEXCOM_IO4	RTS	NPCS1	–	O
FLEXCOM_IO5	–	NPCS2	–	O
FLEXCOM_IO6	–	NPCS3	–	O

63.5 Product Dependencies

63.5.1 I/O Lines

The pins used for interfacing the FLEXCOM are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the desired FLEXCOM pins to their peripheral function. If I/O lines of the FLEXCOM are not used by the application, they can be used for other purposes by the PIO Controller.

63.5.2 Power Management

The peripheral clock is not continuously provided to the FLEXCOM. The programmer must first enable the FLEXCOM Clock in the Power Management Controller (PMC) before using the USART or SPI or TWI.

To enable asynchronous partial wakeup for the FLEXCOM, the PMC must be configured first. The FLEXCOM peripheral clock can be automatically provided depending on the instructions (requests) provided by the FLEXCOM to the PMC.

63.5.3 Interrupt Sources

The FLEXCOM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the FLEXCOM interrupt requires the Interrupt Controller to be programmed first.

63.6 Register Accesses

Register accesses support 8-bit, 16-bit and 32-bit access, allowing, for example, an 8-bit part of a 32-bit register to be written in one access. To do so, the access must be done with the right size at the right address.

8-bit, 16-bit and 32-bit accesses are supported for register accesses. However, a field in a register cannot be partially written (e.g., if a field is bigger than 8 bits, the whole field must be written).

This feature avoids a read-modify-write process if only a small part of the register is to be modified.

63.7 USART Functional Description

63.7.1 Baud Rate Generator

The baud rate generator provides the bit period clock named “baud rate clock” to both the receiver and the transmitter.

Configuring the USCLKS field in FLEX_US_MR selects the baud rate generator clock from one of the following sources:

- the peripheral clock
- a fully programmable generic clock (GCLK) provided by PMC and independent of processor/peripheral clock
- the external clock, available on the SCK pin

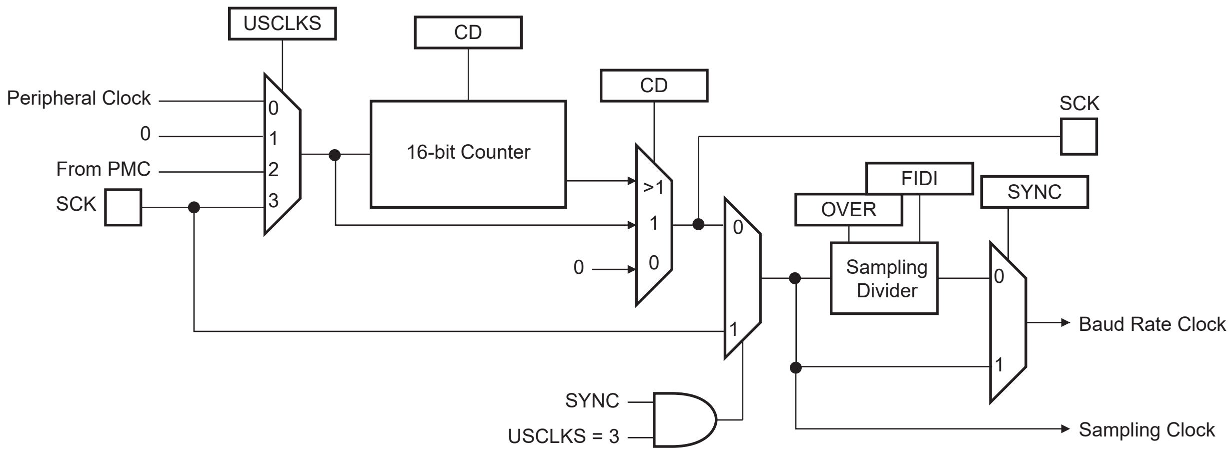
The baud rate generator is based upon a 16-bit divider, which is programmed with the CD field of the Baud Rate Generator register (FLEX_US_BRGR). If a zero is written to CD, the baud rate generator does not generate any clock. If a one is written to CD, the divider is bypassed and becomes inactive.

If the external SCK clock is selected, the duration of the low and high levels of the signal provided on the SCK pin must be longer than a peripheral clock period. The frequency of the signal provided on SCK must be at least three times lower than peripheral clock .

If GCLK is selected, the baud rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the USART transfer. The GCLK frequency must be at least three times lower than peripheral clock frequency.

If GCLK is selected (USCLKS = 2) and the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.

Figure 63-2. Baud Rate Generator



63.7.1.1 Baud Rate in Asynchronous Mode

If the USART is programmed to operate in Asynchronous mode, the selected clock is first divided by CD, which is field-programmed in FLEX_US_BRGR. The resulting clock is provided to the receiver as a sampling clock and then divided by 16 or 8, depending on the programming of FLEX_US_MR.OVER.

If OVER is set, the receiver sampling is eight times higher than the baud rate clock. If OVER is cleared, the sampling is performed at 16 times the baud rate clock.

The baud rate is calculated as per the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{(8(2-\text{OVER})\text{CD})}$$

This gives a maximum baud rate of peripheral clock divided by 8, assuming that peripheral clock is the highest possible clock and that the OVER bit is set.

63.7.1.1.1 Baud Rate Calculation Example

The following table shows calculations of CD to obtain a baud rate at 38,400 bit/s for different source clock frequencies. It also shows the actual resulting baud rate and the error.

Table 63-3. Baud Rate Example (OVER = 0)

Source Clock (MHz)	**Expected Baud Rate** (bit/s)	**Calculation Result**	**CD**	**Actual Baud Rate** (bit/s)	**Error**
3,686,400	38,400	6.00	6	38,400.00	0.00%
4,915,200	38,400	8.00	8	38,400.00	0.00%
5,000,000	38,400	8.14	8	39,062.50	1.70%
7,372,800	38,400	12.00	12	38,400.00	0.00%
8,000,000	38,400	13.02	13	38,461.54	0.16%
12,000,000	38,400	19.53	20	37,500.00	2.40%
12,288,000	38,400	20.00	20	38,400.00	0.00%

...........continued

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
14,318,180	38,400	23.30	23	38,908.10	1.31%
14,745,600	38,400	24.00	24	38,400.00	0.00%
18,432,000	38,400	30.00	30	38,400.00	0.00%
24,000,000	38,400	39.06	39	38,461.54	0.16%
24,576,000	38,400	40.00	40	38,400.00	0.00%
25,000,000	38,400	40.69	40	38,109.76	0.76%
32,000,000	38,400	52.08	52	38,461.54	0.16%
32,768,000	38,400	53.33	53	38,641.51	0.63%
33,000,000	38,400	53.71	54	38,194.44	0.54%
40,000,000	38,400	65.10	65	38,461.54	0.16%
50,000,000	38,400	81.38	81	38,580.25	0.47%

The baud rate is calculated with the following formula:

Baud rate = MCK / CD × 16

The baud rate error is calculated with the following formula. It is not recommended to work with an error higher than 5%.

$$\text{Error} = 1 - \left(\frac{\text{Expected Baud Rate}}{\text{Actual Baud Rate}}\right)$$

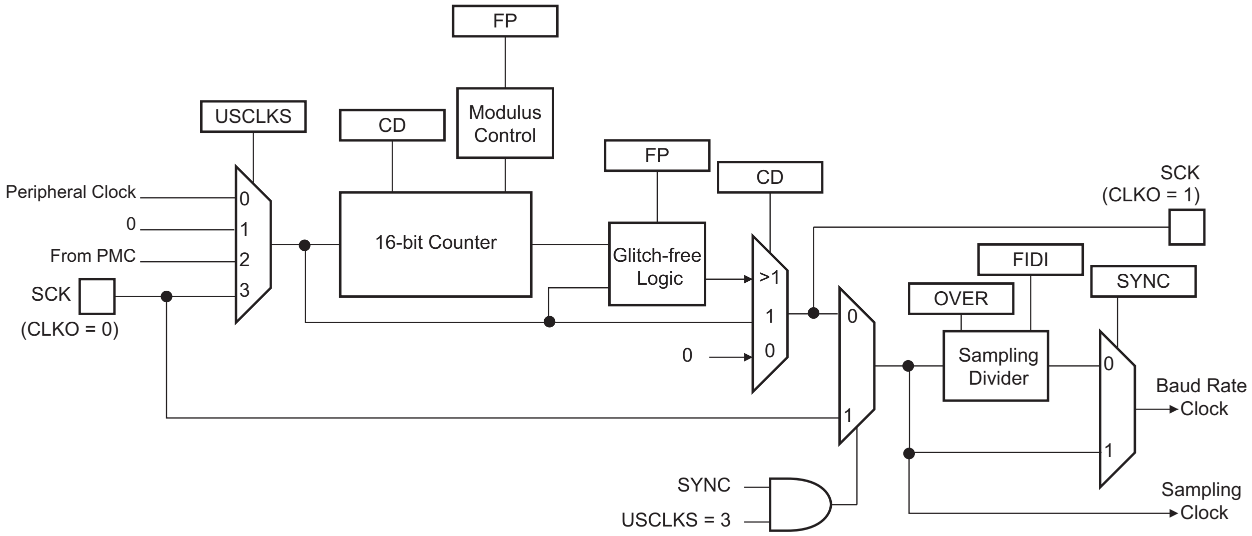
63.7.1.2 Fractional Baud Rate in Asynchronous Mode

The baud rate generator previously defined is subject to the following limitation: the output frequency changes by only integer multiples of the reference frequency. An approach to this problem is to integrate a fractional N clock generator that has a high resolution. The generator architecture is modified to obtain baud rate changes by a fraction of the reference source clock. This fractional part is programmed with the FP field in FLEX_US_BRGR. If FP is not 0, the fractional part is activated. The resolution is one eighth of the clock divider. The fractional baud rate is calculated using the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\left(8(2 - \text{OVER})\left(\text{CD} + \frac{\text{FP}}{8}\right)\right)}$$

The modified architecture is presented in the following figure.

Figure 63-3. Fractional Baud Rate Generator



⚠WARNING When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by "selected clock" period from time to time. The duty cycle depends on the value of the CD field.

63.7.1.3 Baud Rate in Synchronous Mode

If the USART is programmed to operate in Synchronous mode, the selected clock is simply divided by the CD field in FLEX_US_BRGR:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\text{CD}}$$

In Synchronous mode, if the external clock is selected (USCLKS = 3) and CLKO = 0 (Client mode), the clock is provided directly by the signal on the USART SCK pin. No division is active. The value written in FLEX_US_BRGR has no effect. The external clock frequency must be at least three times lower than the system clock. In Synchronous mode host (USCLKS = 0, CLKO = 1), the receive part limits the SCK maximum frequency to $f_{\text{peripheral clock}}/3$.

When either the external clock SCK or the internal clock divided (GCLK) is selected and if the user has to ensure a 50:50 mark/space ratio on the SCK pin, the value programmed in CD must be even. If the peripheral clock is selected and if the value programmed in CD is odd, the baud rate generator ensures a 50:50 duty cycle on the SCK pin.

63.7.1.4 Baud Rate in ISO 7816 Mode

The ISO7816 specification defines the bit rate with the following formula:

$$B = \frac{\text{Di}}{\text{Fi}} \times f$$

where:

- B is the bit rate
- Di is the bit rate adjustment factor
- Fi is the clock frequency division factor
- f is the ISO7816 clock frequency (Hz)

Di is a binary value encoded on a 4-bit field, named DI, as represented in the following table.

Table 63-4. Binary and Decimal Values for Di

DI field	0001	0010	0011	0100	0101	0110	1000	1001
Di (decimal)	1	2	4	8	16	32	12	20

Fi is a binary value encoded on a 4-bit field, named FI, as represented in the following table.

Table 63-5. Binary and Decimal Values for Fi

FI field	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
Fi (decimal)	372	372	558	744	1116	1488	1860	512	768	1024	1536	2048

The following table shows the resulting Fi/Di Ratio, which is the ratio between the ISO7816 clock and the baud rate clock.

Table 63-6. Possible Values for the Fi/Di Ratio

Fi/Di	372	558	744	1116	1488	1806	512	768	1024	1536	2048
1	372	558	744	1116	1488	1860	512	768	1024	1536	2048
2	186	279	372	558	744	930	256	384	512	768	1024
4	93	139.5	186	279	372	465	128	192	256	384	512
8	46.5	69.75	93	139.5	186	232.5	64	96	128	192	256
16	23.25	34.87	46.5	69.75	93	116.2	32	48	64	96	128
32	11.62	17.43	23.25	34.87	46.5	58.13	16	24	32	48	64
12	31	46.5	62	93	124	155	42.66	64	85.33	128	170.6
20	18.6	27.9	37.2	55.8	74.4	93	25.6	38.4	51.2	76.8	102.4

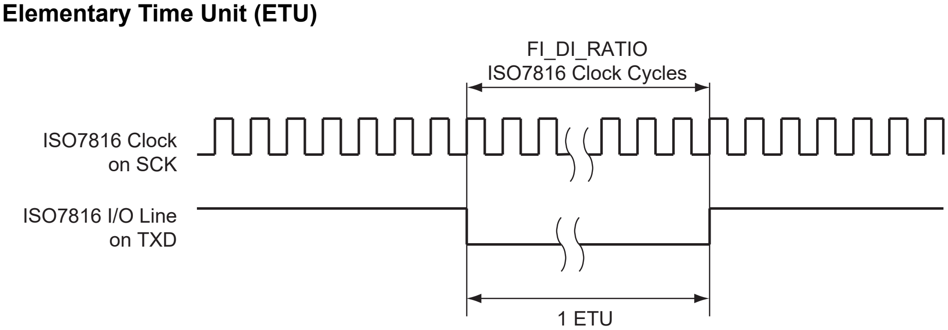
If the USART is configured in ISO7816 mode, the clock selected by the USCLKS field in FLEX_US_MR is first divided by the value programmed in field CD field in FLEX_US_BRGR. The resulting clock can be provided to the SCK pin to feed the smart card clock inputs. This means that FLEX_US_MR.CLKO can be set.

This clock is then divided by the value programmed in the FI_DI_RATIO field in the FI DI Ratio register (FLEX_US_FIDI). This is performed by the Sampling Divider, which performs a division by up to 65535 in ISO7816 mode. The noninteger values of the Fi/Di Ratio are not supported and the user must program the FI_DI_RATIO field to a value as close as possible to the expected value.

The FI_DI_RATIO field resets to the value 0x174 (372 in decimal) and is the most common divider between the ISO7816 clock and the bit rate (Fi = 372, Di = 1).

The following figure shows the relation between the Elementary Time Unit, corresponding to a bit time, and the ISO 7816 clock.

Figure 63-4. Elementary Time Unit (ETU)



63.7.2 Receiver and Transmitter Control

After reset, the receiver is disabled. The user must enable the receiver by setting the RXEN bit in the USART Control register (FLEX_US_CR). However, the receiver registers can be programmed before the receiver clock is enabled.

After reset, the transmitter is disabled. The user must enable it by setting the TXEN bit in FLEX_US_CR. However, the transmitter registers can be programmed before being enabled.

The receiver and the transmitter can be enabled together or independently.

At any time, the software can perform a reset on the receiver or the transmitter of the USART by setting the corresponding bit, RSTRX and RSTTX respectively, in FLEX_US_CR. The software resets clear the status flag and reset internal state machines but the user interface configuration registers hold the value configured prior to software reset. Regardless of what the receiver or the transmitter is performing, the communication is immediately stopped.

The user can also independently disable the receiver or the transmitter by setting RXDIS and TXDIS respectively in FLEX_US_CR. If the receiver is disabled during a character reception, the USART waits until the end of reception of the current character, then the reception is stopped. If the transmitter is disabled while it is operating, the USART waits the end of transmission of both the current character and character being stored in the USART Transmit Holding register (FLEX_US_THR). If a timeguard is programmed, it is handled normally.

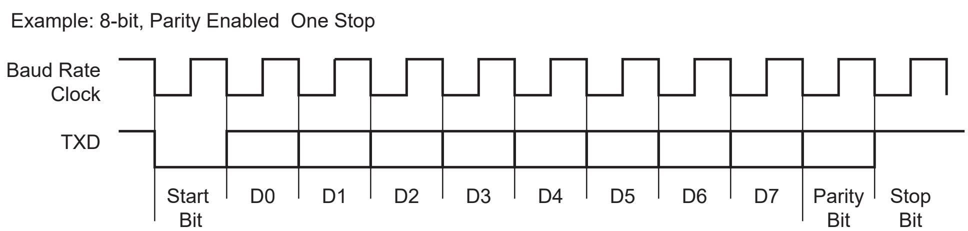
63.7.3 Synchronous and Asynchronous Modes

63.7.3.1 Transmitter Operations

The transmitter performs the same in both Synchronous and Asynchronous operating modes (SYNC = 0 or SYNC = 1). One start bit, up to 9 data bits, 1 optional parity bit and up to 2 stop bits are successively shifted out on the TXD pin at each falling edge of the programmed serial clock.

The number of data bits is selected by the CHRL field and the MODE9 bit in FLEX_US_MR. Nine bits are selected by setting the MODE9 bit regardless of the CHRL field. The parity bit is set according to the PAR field in FLEX_US_MR. The even, odd, space, marked or none parity bit can be configured. The MSBF bit in FLEX_US_MR configures which data bit is sent first. If written to 1, the most significant bit is sent first. If written to 0, the less significant bit is sent first. The number of stop bits is selected by the NBSTOP field in FLEX_US_MR. The 1.5 stop bit is supported in Asynchronous mode only.

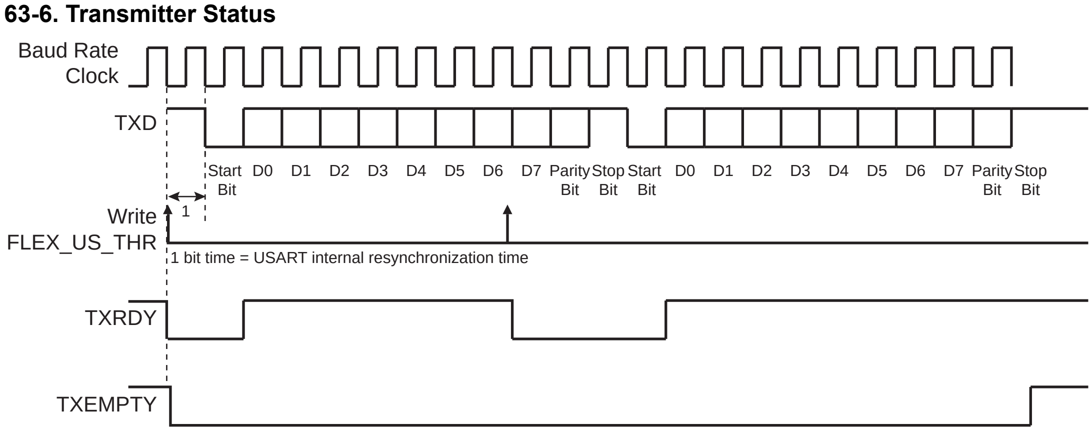
Figure 63-5. Character Transmit



The characters are sent by writing in FLEX_US_THR. The transmitter reports two status bits in the USART Channel Status register (FLEX_US_CSR): TXRDY (Transmitter Ready), which indicates that FLEX_US_THR is empty and TXEMPTY, which indicates that all the characters written in FLEX_US_THR have been processed. When the current character processing is completed, the last character written in FLEX_US_THR is transferred into the shift register of the transmitter and FLEX_US_THR is emptied, thus TXRDY rises.

Both TXRDY and TXEMPTY bits are low when the transmitter is disabled. Writing a character in FLEX_US_THR while TXRDY is low has no effect and the written character is lost.

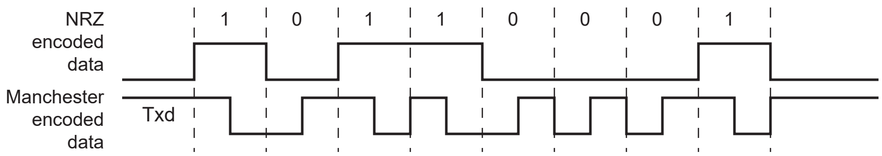
Figure 63-6. Transmitter Status



63.7.3.2 Manchester Encoder

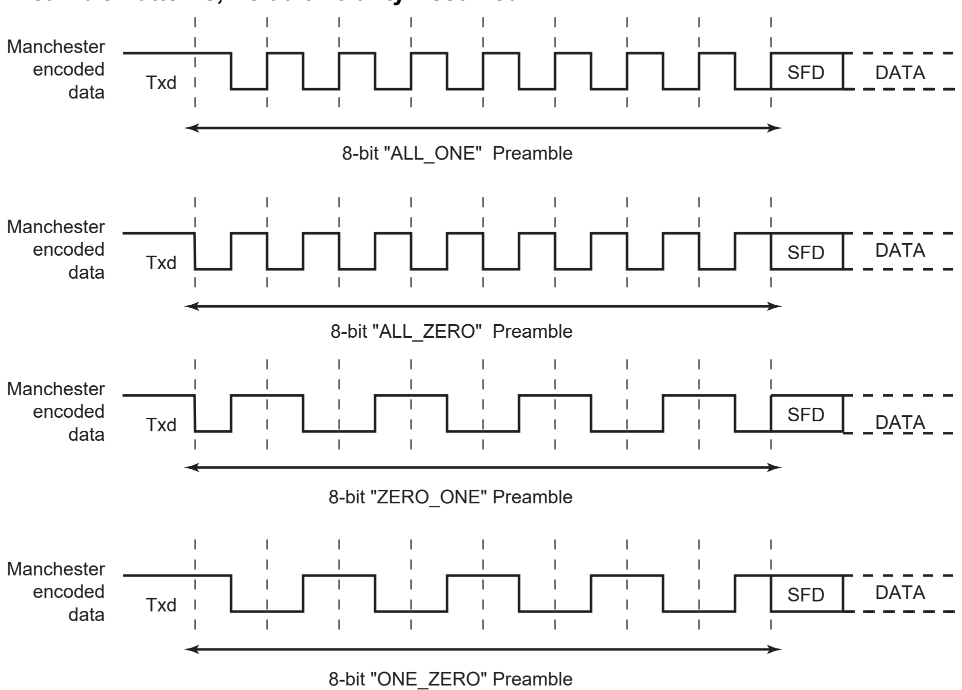
When the Manchester encoder is in use, characters transmitted through the USART are encoded based on biphase Manchester II format. To enable this mode, set the FLEX_US_MR.MAN bit to 1. Depending on polarity configuration, a logic level (zero or one), is transmitted as a coded signal one-to-zero or zero-to-one. Thus, a transition always occurs at the midpoint of each bit time. It consumes more bandwidth than the original NRZ signal (2x) but the receiver has more error control since the expected input must show a change at the center of a bit cell. An example of Manchester encoded sequence is: the byte 0xB1 or 10110001 encodes to 10 01 10 10 01 01 01 10, assuming the default polarity of the encoder. The following figure illustrates this coding scheme.

Figure 63-7. NRZ to Manchester Encoding



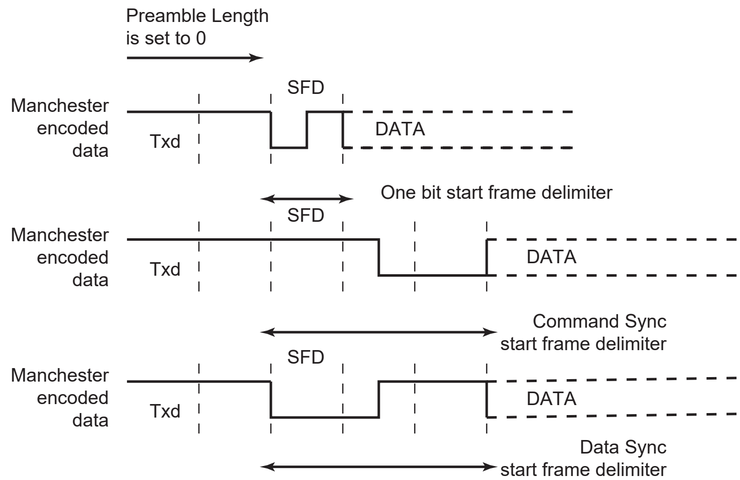
The Manchester encoded character can also be encapsulated by adding both a configurable preamble and a start frame delimiter pattern. Depending on the configuration, the preamble is a training sequence, composed of a predefined pattern with a programmable length from 1 to 15 bit times. If the preamble length is set to 0, the preamble waveform is not generated prior to any character. The preamble pattern is chosen among the following sequences: ALL_ONE, ALL_ZERO, ONE_ZERO or ZERO_ONE, writing the FLEX_US_MAN.TX_PP field. The TX_PL field is used to configure the preamble length. The following figure illustrates and defines the valid patterns. To improve flexibility, the encoding scheme can be configured using the FLEX_US_MAN.TX_MPOL bit. If the TX_MPOL bit is set to zero (default), a logic zero is encoded with a zero-to-one transition and a logic one is encoded with a one-to-zero transition. If the TX_MPOL bit is set to one, a logic one is encoded with a one-to-zero transition and a logic zero is encoded with a zero-to-one transition.

Figure 63-8. Preamble Patterns, Default Polarity Assumed



A start frame delimiter is to be configured using the FLEX_US_MR.ONEBIT bit. It consists of a user-defined pattern that indicates the beginning of a valid data. The following figure illustrates these patterns. If the start frame delimiter, also known as the start bit, is one bit, (ONEBIT = 1), a logic zero is Manchester encoded and indicates that a new character is being sent serially on the line. If the start frame delimiter is a synchronization pattern also referred to as sync (ONEBIT = 0), a sequence of three bit times is sent serially on the line to indicate the start of a new character. The sync waveform is in itself an invalid Manchester waveform as the transition occurs at the middle of the second bit time. Two distinct sync patterns are used: the command sync and the data sync. The command sync has a logic one level for one and a half bit times, then a transition to logic zero for the second one and a half bit times. If the FLEX_US_MR.MODSYNC bit is set to 1, the next character is a command. If it is set to 0, the next character is a data. When direct memory access is used, the MODSYNC bit can be immediately updated with a modified character located in memory. To enable this mode, the FLEX_US_MR.VAR_SYNC bit must be set. In this case, the FLEX_US_MR.MODSYNC bit is bypassed and the sync configuration is held in the FLEX_US_THR.TXSYNH bit. The USART character format is modified and includes sync information.

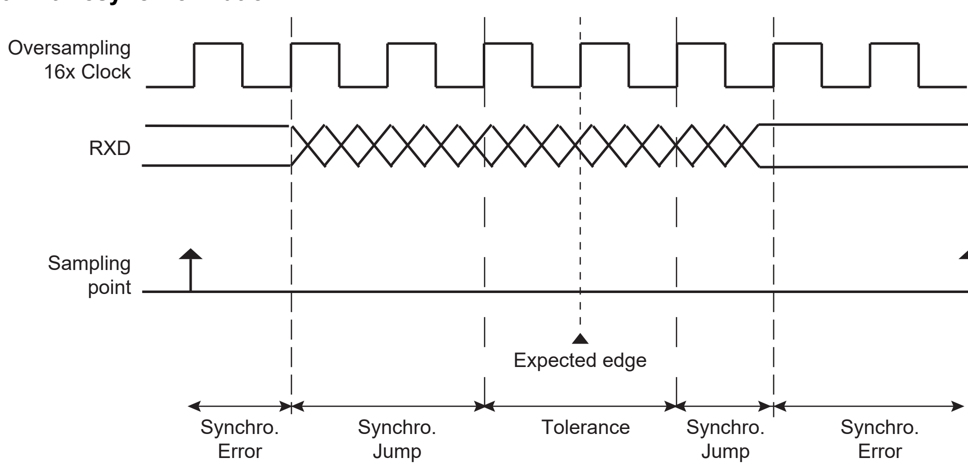
Figure 63-9. Start Frame Delimiter



63.7.3.2.1 Drift Compensation

Drift compensation is available only in 16X Oversampling mode. An hardware recovery system allows a larger clock drift. To enable the hardware system, the bit in the FLEX_US_MAN register must be set. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective actions is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

Figure 63-10. Bit Resynchronization



63.7.3.3 Asynchronous Receiver

If the USART is programmed in Asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the FLEX_US_MR.OVER bit.

The receiver samples the RXD line. If the line is sampled during one half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

If the oversampling is 16 (OVER = 0), a start is detected at the eighth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 16 oversampling clock cycles. If the oversampling is 8 (OVER =

1), a start bit is detected at the fourth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 8 oversampling clock cycles.

The number of data bits, first bit sent and Parity mode are selected by the same fields and bits as the transmitter, i.e., respectively CHRL, MODE9, MSBF and PAR. For the synchronization mechanism only, the number of stop bits has no effect on the receiver as it considers only one stop bit, regardless of the NBSTOP field, so that resynchronization between the receiver and the transmitter can occur. Moreover, as soon as the stop bit is sampled, the receiver starts looking for a new start bit so that resynchronization can also be accomplished when the transmitter is operating with one stop bit.

The following figures illustrate start detection and character reception when USART operates in Asynchronous mode.

Figure 63-11. Asynchronous Start Detection

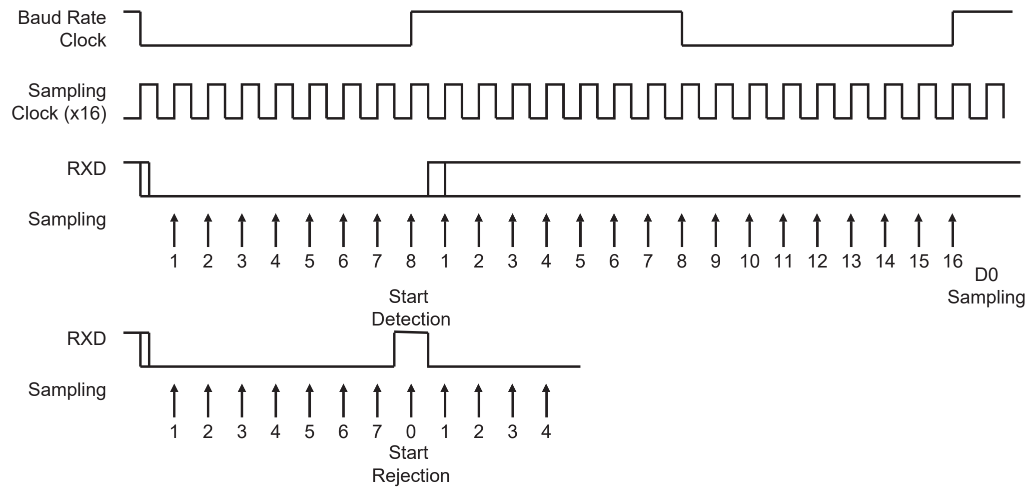
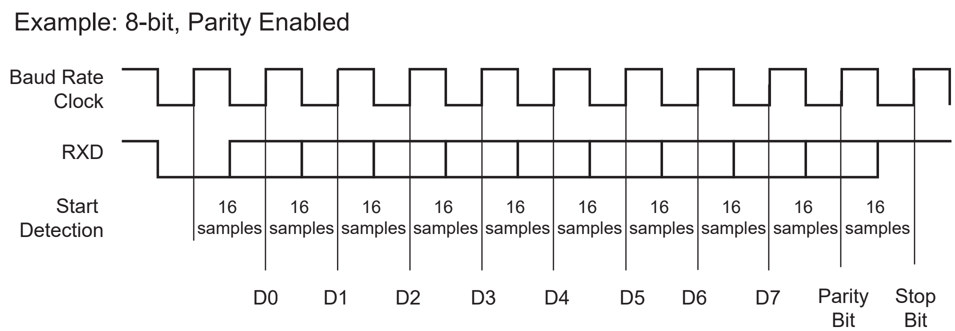


Figure 63-12. Asynchronous Character Reception



63.7.3.4 Manchester Decoder

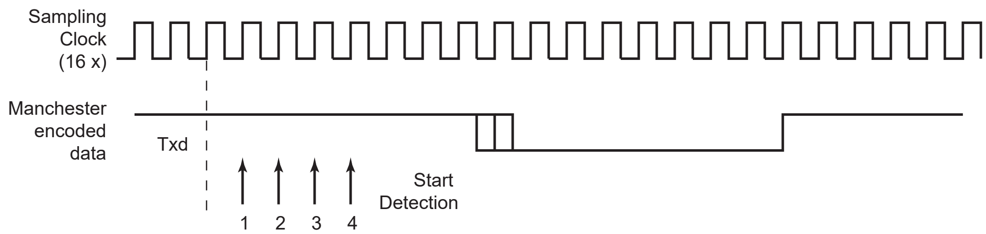
When the FLEX_US_MR.MAN bit is set, the Manchester decoder is enabled. The decoder performs both preamble and start frame delimiter detection. One input line is dedicated to Manchester encoded input data.

An optional preamble sequence can be defined. Its length is user-defined and totally independent of the transmitter side. Use the FLEX_US_MAN.RX_PL field to configure the length of the preamble sequence. If the length is set to 0, no preamble is detected and the function is disabled. In addition, the polarity of the input stream is programmable with the FLEX_US_MAN.RX_MPOL bit. Depending on the desired application, the preamble pattern matching is to be defined via the FLEX_US_MAN.RX_PP field. See figure Preamble Patterns, Default Polarity Assumed for available preamble patterns.

Unlike preamble, the start frame delimiter is shared between Manchester Encoder and Decoder. So, if ONEBIT bit = 1, only a zero encoded Manchester can be detected as a valid start frame delimiter. If ONEBIT = 0, only a sync pattern is detected as a valid start frame delimiter. Decoder operates by detecting transition on incoming stream. If RXD is sampled during one quarter of a bit time to zero, a start bit is detected. See the following figure. The sample pulse rejection mechanism applies.

The FLEX_US_MAN.RXIDLEV bit informs the USART of the receiver line idle state value (receiver line inactive). The user must define RXIDLEV to ensure reliable synchronization. By default, RXIDLEV is set to one (receiver line is at level 1 when there is no activity).

Figure 63-13. Asynchronous Start Bit Detection



The receiver is activated and starts preamble and frame delimiter detection, sampling the data at one quarter and then three quarters. If a valid preamble pattern or start frame delimiter is detected, the receiver continues decoding with the same synchronization. If the stream does not match a valid pattern or a valid start frame delimiter, the receiver resynchronizes on the next valid edge.The minimum time threshold to estimate the bit value is three quarters of a bit time.

If a valid preamble (if used) followed with a valid start frame delimiter is detected, the incoming stream is decoded into NRZ data and passed to USART for processing. The following figure illustrates Manchester pattern mismatch. When incoming data stream is passed to the USART, the receiver is also able to detect Manchester code violation. A code violation is a lack of transition in the middle of a bit cell. In this case, the MANE flag in FLEX_US_CSR is raised. It is cleared by writing a one to FLEX_US_CR.RSTSTA. See figure "Manchester Error Flag" below for an example of Manchester error detection during the data phase.

Figure 63-14. Preamble Pattern Mismatch

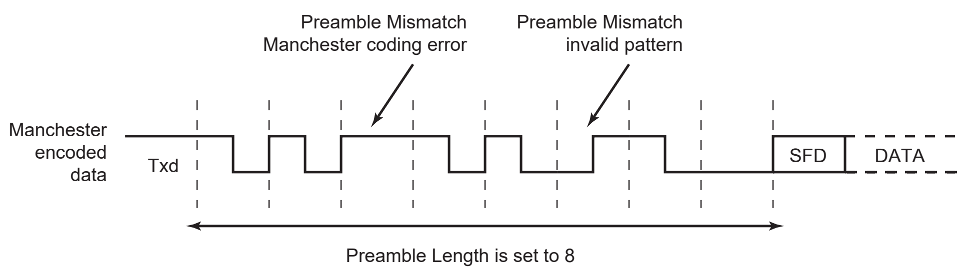
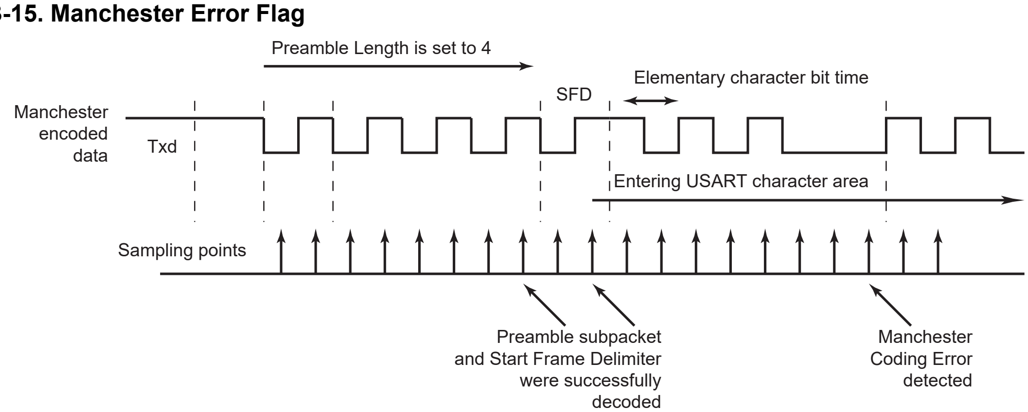


Figure 63-15. Manchester Error Flag



When the start frame delimiter is a sync pattern (ONEBIT = 0), both command and data delimiter are supported. If a valid sync is detected, the received character is written as RXCHR field in the Receive Holding register (FLEX_US_RHR) and the RXSYNH is updated. RXCHR is set to 1 when the received character is a command, and it is set to 0 if the received character is a data. This mechanism alleviates and simplifies the direct memory access as the character contains its own sync field in the same register.

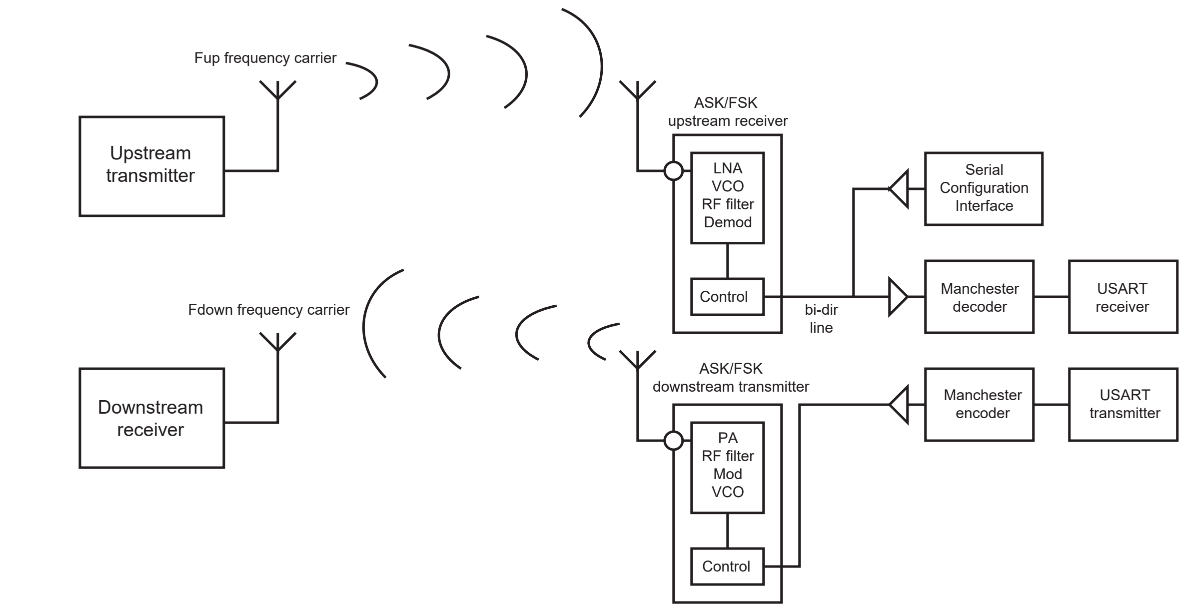
As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

63.7.3.5 Radio Interface: Manchester Encoded USART Application

This section describes low data rate RF transmission systems and their integration with a Manchester encoded USART. These systems are based on transmitter and receiver ICs that support ASK and FSK modulation schemes.

The goal is to perform full-duplex radio transmission of characters using two different frequency carriers. See configuration in the following figure.

Figure 63-16. Manchester Encoded Characters RF Transmission



The USART peripheral is configured as a Manchester encoder/decoder. Looking at the downstream communication channel, Manchester encoded characters are serially sent to the RF transmitter. This may also include a user defined preamble and a start frame delimiter. Mostly, preamble is used in the RF receiver to distinguish between a valid data from a transmitter and signals due to noise. The Manchester stream is then modulated. See the following figure for an example of ASK modulation scheme. When a logic one is sent to the ASK modulator, the power amplifier, referred to as PA, is enabled and transmits an RF signal at downstream frequency. When a logic zero is transmitted, the RF signal is turned off. If the FSK modulator is activated, two different frequencies are used to transmit data. When a logic 1 is sent, the modulator outputs an RF signal at frequency F0 and switches to F1 if the data sent is a 0. See figure "FSK Modulator Output" below.

From the receiver side, another carrier frequency is used. The RF receiver performs a bit check operation examining demodulated data stream. If a valid pattern is detected, the receiver switches to Receiving mode. The demodulated stream is sent to the Manchester decoder. Because of bit checking inside RF IC, the data transferred to the microcontroller is reduced by a user-defined number of bits. The Manchester preamble length is to be defined in accordance with the RF IC configuration.

Figure 63-17. ASK Modulator Output

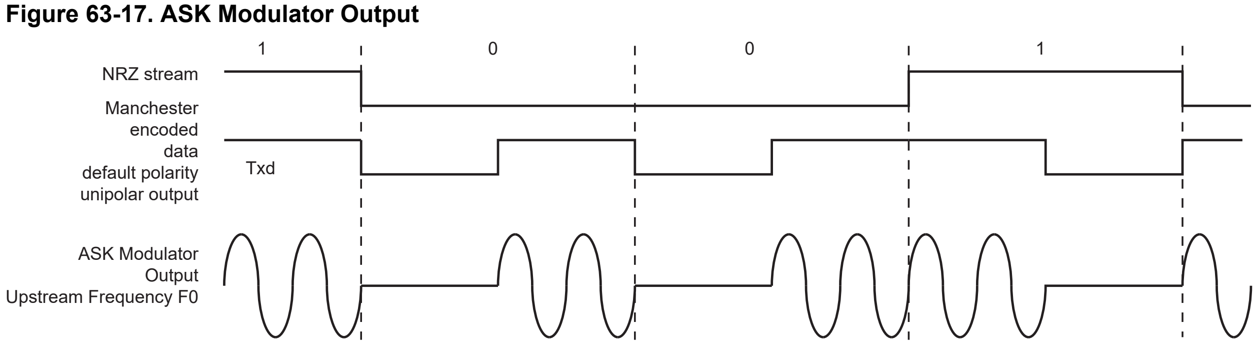
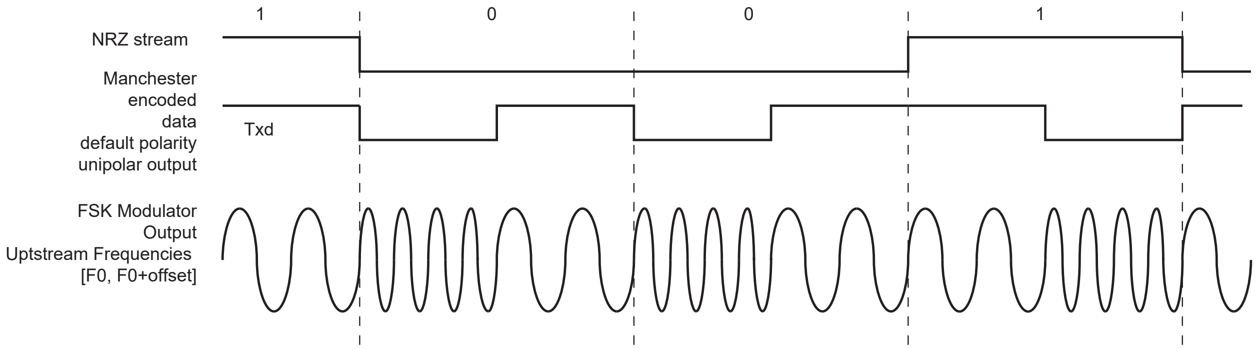


Figure 63-18. FSK Modulator Output



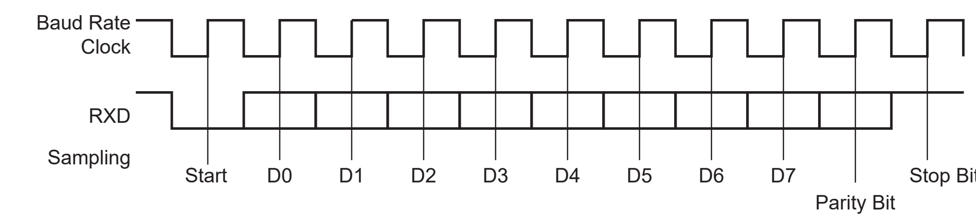
63.7.3.6 Synchronous Receiver

In Synchronous mode (SYNC = 1), the receiver samples the RXD signal on each rising edge of the baud rate clock. If a low level is detected, it is considered as a start. All data bits, the parity bit and the stop bits are sampled and the receiver waits for the next start bit. Synchronous mode operations provide a high-speed transfer capability.

Configuration fields and bits are the same as in Asynchronous mode.

The following figure illustrates a character reception in Synchronous mode.

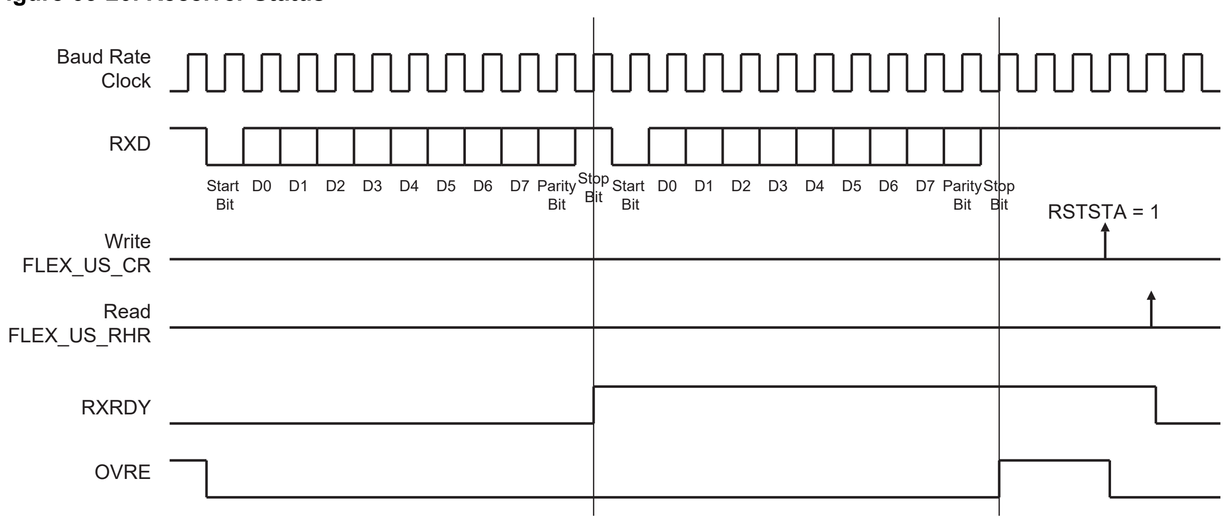
Figure 63-19. Synchronous Mode Character Reception



63.7.3.7 Receiver Operations

When a character reception is completed, it is transferred to the Receive Holding register (FLEX_US_RHR) and the FLEX_US_CSR.RXRDY bit is raised. If a character is completed while the RXRDY is set, the Overrun Error (OVRE) bit is set. The last character is transferred into FLEX_US_RHR and overwrites the previous one. The OVRE bit is cleared by writing a one to Reset Status bit FLEX_US_CR.RSTSTA.

Figure 63-20. Receiver Status



63.7.3.8 Parity

The USART supports five parity modes that are selected by writing to the FLEX_US_MR.PAR field. The PAR field also enables the Multidrop mode (see Multidrop Mode). Even and odd parity bit generation and error detection are supported.

If even parity is selected, the parity generator of the transmitter drives the parity bit to 0 if a number of 1s in the character data bit is even, and to 1 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If odd parity is selected, the parity generator of the transmitter drives the parity bit to 1 if a number of 1s in the character data bit is even, and to 0 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If the mark parity is used, the parity generator of the transmitter drives the parity bit to 1 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 0. If the space parity is used, the parity generator of the transmitter drives the parity bit to 0 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 1. If parity is disabled, the transmitter does not generate any parity bit and the receiver does not report any parity error.

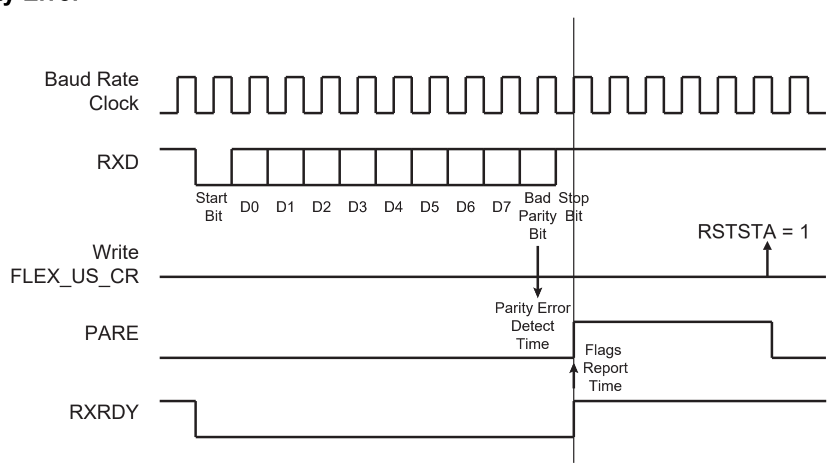
The following table shows an example of the parity bit for the character 0x41 (character ASCII “A”) depending on the configuration of the USART. Because there are two bits set to 1 in the character value, the parity bit is set to 1 when the parity is odd, or configured to 0 when the parity is even.

Table 63-7. Parity Bit Examples

Character	Hexadecimal	Binary	Parity Bit	Parity Mode
A	0x41	0100 0001	1	Odd
A	0x41	0100 0001	0	Even
A	0x41	0100 0001	1	Mark
A	0x41	0100 0001	0	Space
A	0x41	0100 0001	None	None

When the receiver detects a parity error, it sets the Parity Error bit FLEX_US_CSR.PARE. The PARE bit can be cleared by writing a one to the FLEX_US_CR.RSTSTA bit. The following figure illustrates the parity bit status setting and clearing.

Figure 63-21. Parity Error



63.7.3.9 Multidrop Mode

If the value 0x6 or 0x07 is written to the FLEX_US_MR.PAR field, the USART runs in Multidrop mode. This mode differentiates the data characters and the address characters. Data are transmitted with the parity bit to 0 and addresses are transmitted with the parity bit to 1.

If the USART is configured in Multidrop mode, the receiver sets the PARE parity error bit when the parity bit is high and the transmitter is able to send a character with the parity bit high when a one is written to the FLEX_US_CR.SENDA bit.

To handle parity error, the PARE bit is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

The transmitter sends an address byte (parity bit set) when the FLEX_US_CR.SENDA bit is written to 1. In this case, the next byte written to FLEX_US_THR is transmitted as an address. Any character written in FLEX_US_THR when the SENDA command is not written is transmitted normally with parity to 0.

63.7.3.10 Transmitter Timeguard

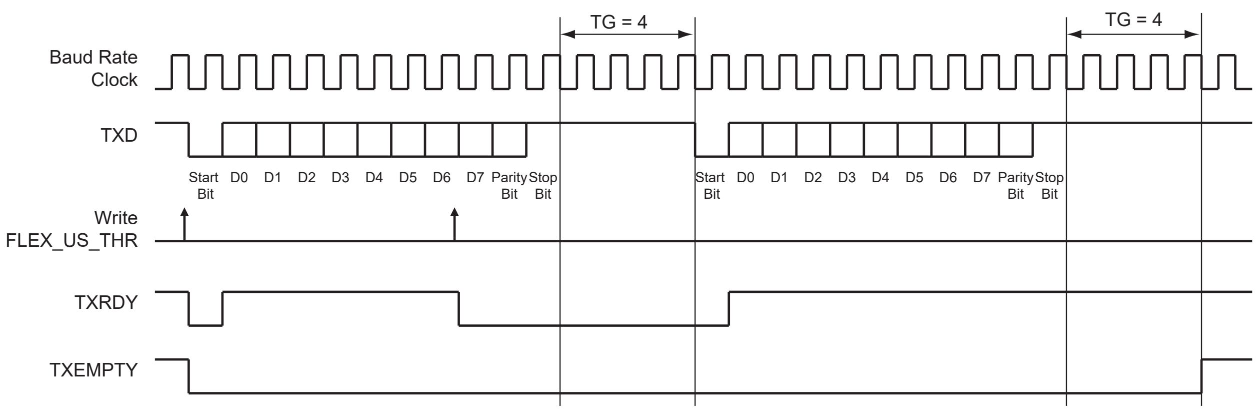
The timeguard feature enables the USART interface with slow remote devices.

The timeguard function enables the transmitter to insert an idle state on the TXD line between two characters. This idle state actually acts as a long stop bit.

The duration of the idle state is programmed in the TG field of the Transmitter Timeguard register (FLEX_US_TTGR). When this field is written to zero, no timeguard is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted byte during the number of bit periods programmed in TG in addition to the number of stop bits.

As illustrated in the following figure, the behavior of the TXRDY and TXEMPTY status bits is modified by the programming of a timeguard. TXRDY rises only when the start bit of the next character is sent, and thus remains to 0 during the timeguard transmission if a character has been written in FLEX_US_THR. TXEMPTY remains low until the timeguard transmission is completed as the timeguard is part of the current character being transmitted.

Figure 63-22. Timeguard Operations



The following table indicates the maximum length of a timeguard period that the transmitter can handle in relation to the function of the baud rate.

Table 63-8. Maximum Timeguard Length Depending on Baud Rate

Baud Rate (bit/s)	Bit Time (µs)	Timeguard (ms)
1,200	833	212.50
9,600	104	26.56
14,400	69.4	17.71
19,200	52.1	13.28
28,800	34.7	8.85
38,400	26	6.63
56,000	17.9	4.55
57,600	17.4	4.43
115,200	8.7	2.21

63.7.3.11 Receiver Timeout

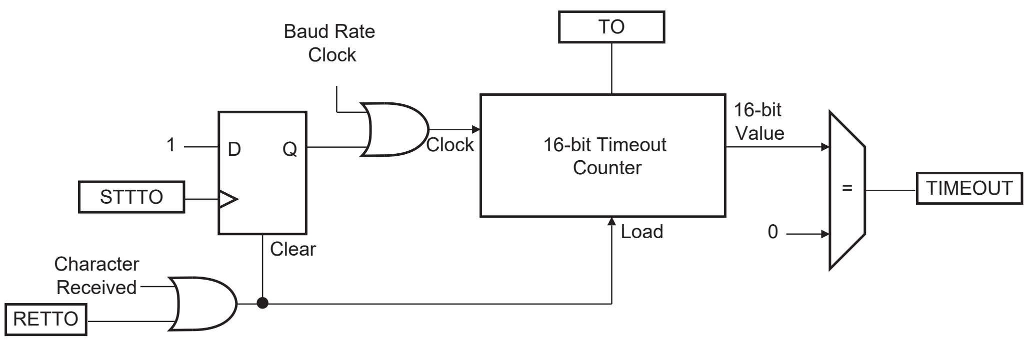
The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver an end of frame.

The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout register (FLEX_US_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 16-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises. Then, the user can either:

- Stop the counter clock until a new character is received. This is performed by writing a ‘1’ to FLEX_US_CR.STTTO. In this case, the idle state on RXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and enables waiting for the next idle state on RXD after a frame is received.
- Obtain an interrupt while no character is received. This is performed by writing a ‘1’ to FLEX_US_CR.RETTO. In this case, the counter starts counting down immediately from the value TO. This generates a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

The following figure shows the block diagram of the Receiver Timeout feature.

Figure 63-23. Receiver Timeout Block Diagram



The following table gives the maximum timeout period for some standard baud rates.

Table 63-9. Maximum Timeout Period

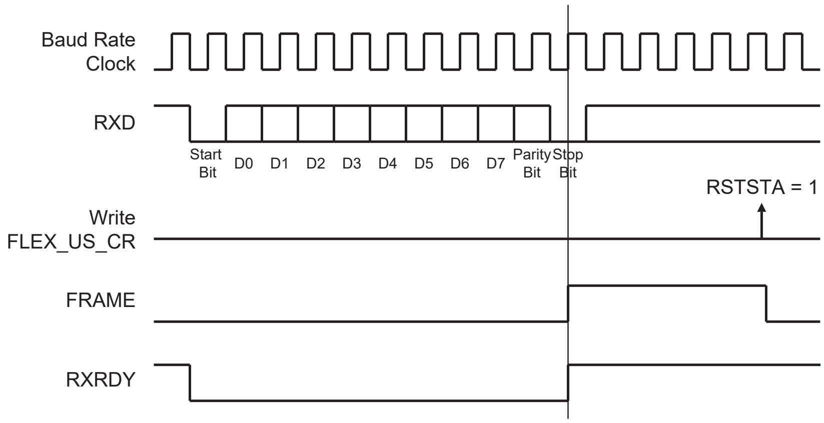
Baud Rate (bit/s)	Bit Time (µs)	Timeout (ms)
600	1,667	109,225
1,200	833	54,613
2,400	417	27,306
4,800	208	13,653
9,600	104	6,827
14,400	69	4,551
19,200	52	3,413
28,800	35	2,276
38,400	26	1,704
56,000	18	1,170
57,600	17	1,138
200,000	5	328

63.7.3.12 Framing Error

The receiver is capable of detecting framing errors. A framing error happens when the stop bit of a received character is detected at level 0. This can occur if the receiver and the transmitter are fully desynchronized.

A framing error is reported on the FLEX_US_CSR.FRAME bit. The FRAME bit is asserted in the middle of the stop bit as soon as the framing error is detected. It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

Figure 63-24. Framing Error Status



63.7.3.13 Transmit Break

The user can request the transmitter to generate a break condition on the TXD line. A break condition drives the TXD line low during at least one complete character. It appears the same as a 0x00 character sent with the parity and the stop bits to 0. However, the transmitter holds the TXD line at least during one character until the user requests the break condition to be removed.

A break is transmitted by setting the FLEX_US_CR.STTBRK bit. This can be done at any time, either while the transmitter is empty (no character in either the shift register or in FLEX_US_THR) or when a character is being transmitted. If a break is requested while a character is being shifted out, the character is first completed before the TXD line is held low.

Once the Start Break command is requested, further Start Break commands are ignored until the end of the break is completed.

The break condition is removed by setting the FLEX_US_CR.STPBRK bit. If the Stop Break command is requested before the end of the minimum break duration (one character, including start, data, parity and stop bits), the transmitter ensures that the break condition completes.

The transmitter considers the break as though it is a character, i.e., the Start Break and Stop Break commands are processed only if the FLEX_US_CSR.TXRDY bit = 1 and the start of the break condition clears the TXRDY and TXEMPTY bits as if a character was processed.

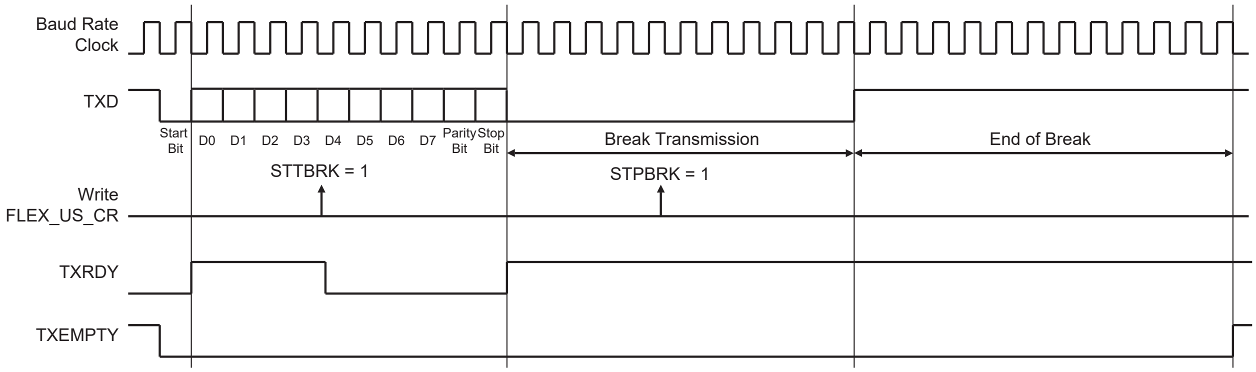
Setting both the FLEX_US_CR.STTBRK and FLEX_US_CR.STPBRK bits can lead to an unpredictable result. All Stop Break commands requested without a previous Start Break command are ignored. A byte written into the Transmit Holding register while a break is pending, but not started, is ignored.

After the break condition, the transmitter returns the TXD line to 1 for a minimum of 12 bit times. Thus, the transmitter ensures that the remote receiver detects correctly the end of break and the start of the next character. If the timeguard is programmed with a value higher than 12, the TXD line is held high for the timeguard period.

After holding the TXD line for this period, the transmitter resumes normal operations.

The following figure illustrates the effect of both the Start Break (STTBRK) and Stop Break (STPBRK) commands on the TXD line.

Figure 63-25. Break Transmission



63.7.3.14 Receive Break

The receiver detects a break condition when all data, parity and stop bits are low. This corresponds to detecting a framing error with data to 0x00, but FRAME remains low.

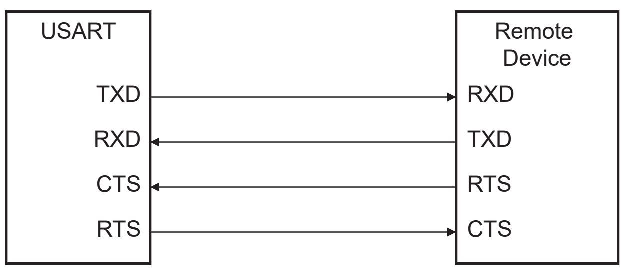
When the low stop bit is detected, the receiver asserts the FLEX_US_CSR.RXBRK bit. FLEX_US_CSR.RXBRK may be cleared by setting the FLEX_US_CR.RSTSTA bit.

An end of receive break is detected by a high level for at least 2/16ths of a bit period in Asynchronous operating mode or one sample at high level in Synchronous operating mode. The end of break detection also asserts the RXBRK bit.

63.7.3.15 Hardware Handshaking

The USART features a hardware handshaking out-of-band flow control. The RTS and CTS pins are used to connect with the remote device, as shown in the following figure.

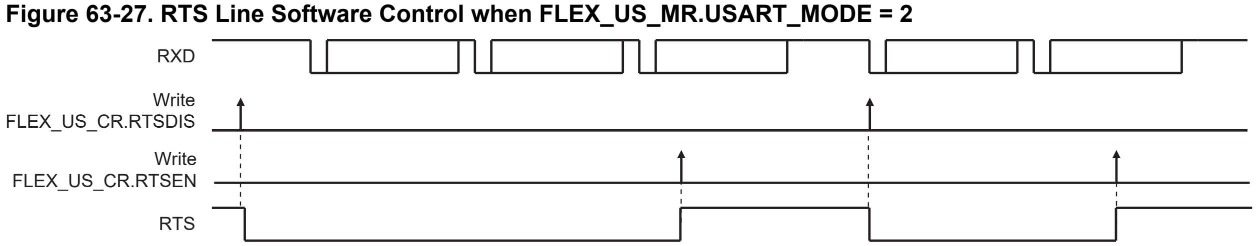
Figure 63-26. Connection with a Remote Device for Hardware Handshaking



Setting the USART to operate with hardware handshaking is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x2.

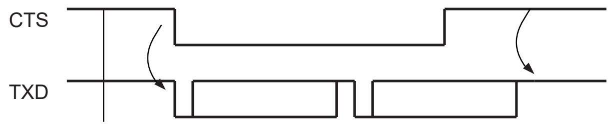
The USART behavior when hardware handshaking is enabled is the same as the behavior in standard Synchronous or Asynchronous mode, except that the receiver drives the RTS pin as described below and the level on the CTS pin modifies the behavior of the transmitter as described below. Using this mode requires using the DMAC channel for reception. The transmitter can handle hardware handshaking in any case.

Figure 63-27. RTS Line Software Control when FLEX_US_MR.USART_MODE = 2



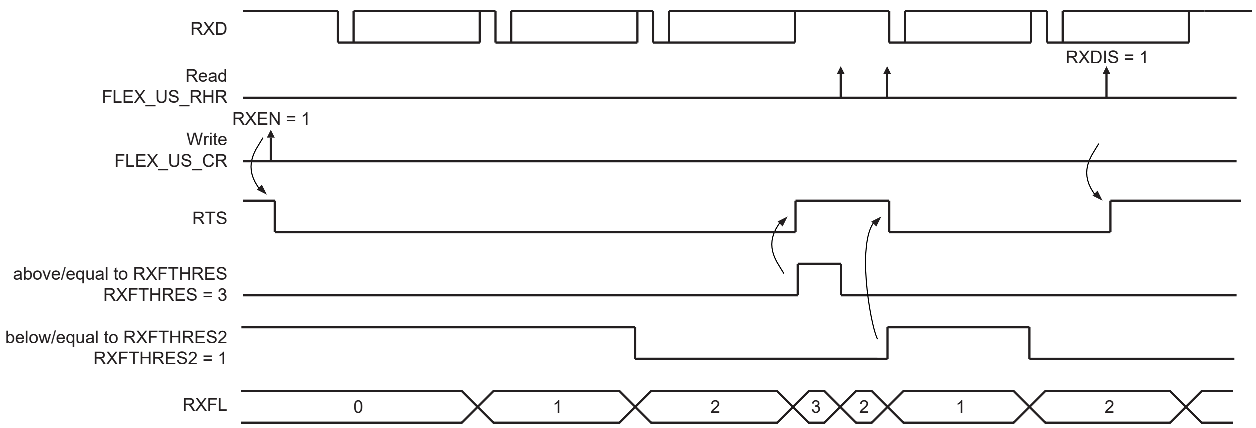
The following figure shows how the transmitter operates if hardware handshaking is enabled. The CTS pin disables the transmitter. If a character is being processed, the transmitter is disabled only after the completion of the current character and transmission of the next character happens as soon as the pin CTS falls.

Figure 63-28. Transmitter Behavior when Operating with Hardware Handshaking



If USART FIFOs are enabled (bit FLEX_US_CR.FIFOEN), the RTS pin can be controlled by the USART Receive FIFO thresholds. The RTS pin control through Receive FIFO thresholds can be activated with the FLEX_US_FMR.FRTSC bit. Once activated, the RTS pin will be controlled by Receive FIFO thresholds, set to level 1 each time RXFTHRES is reached and set to level ‘0’ each time RXFTHRES2 is reached (and RXFTHRES is not reached).

Figure 63-29. Receiver Behavior When FIFO Enabled and FRTSC Set to ‘1’



Note: In this mode, RXFTHRES must be > RXFTHRES2.

63.7.4 ISO7816 Mode

The USART features an ISO7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO7816 link. Both T = 0 and T = 1 protocols defined by the ISO7816 specification are supported.

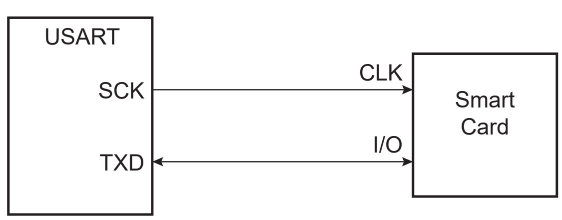
Setting the USART in ISO7816 mode is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x4 for protocol T = 0 and to the value 0x6 for protocol T = 1.

63.7.4.1 ISO7816 Mode Overview

The ISO7816 is a half-duplex communication on only one bidirectional line. The baud rate is determined by a division of the clock provided to the remote device (see figure in section Baud Rate Generator).

The USART connects to a smart card as shown in the following figure. The TXD line becomes bidirectional and the baud rate generator feeds the ISO7816 clock on the SCK pin. As the TXD pin becomes bidirectional, its output remains driven by the output of the transmitter but only when the transmitter is active while its input is directed to the input of the receiver. The USART is considered as the host of the communication as it generates the clock.

Figure 63-30. Connection of a Smart Card to the USART



When operating in ISO7816, either in T = 0 or T = 1 modes, the character format is partially predefined. The configuration is forced to 8 data bits, and 1 or 2 stop bits, regardless of the values programmed in the CHRL, MODE9 and CHMODE fields. MSBF can be used to transmit LSB or MSB first. The bit INVDATA can be used to transmit in Normal or Inverse mode.

The USART cannot operate concurrently in both Receiver and Transmitter modes as the communication is unidirectional at a time. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO7816 mode may lead to unpredictable results.

The ISO7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value.

63.7.4.2 Protocol T = 0

In T = 0 protocol, a character is made up of 1 start bit, 8 data bits, 1 parity bit and 1 guard time, which lasts two bit times. The transmitter shifts out the bits and does not drive the I/O line during the guard time.

If no parity error is detected, the I/O line remains at 1 during the guard time and the transmitter can continue with the transmission of the next character, as shown in the following figure.

If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in figure "T = 0 Protocol with Parity Error" below. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time which lasts 1 bit time.

When the USART is the receiver and it detects an error, it does not load the erroneous character in the Receive Holding register (FLEX_US_RHR). It appropriately sets the PARE bit in the Status register (FLEX_US_CSR) so that the software can handle the error.

Figure 63-31. T = 0 Protocol without Parity Error

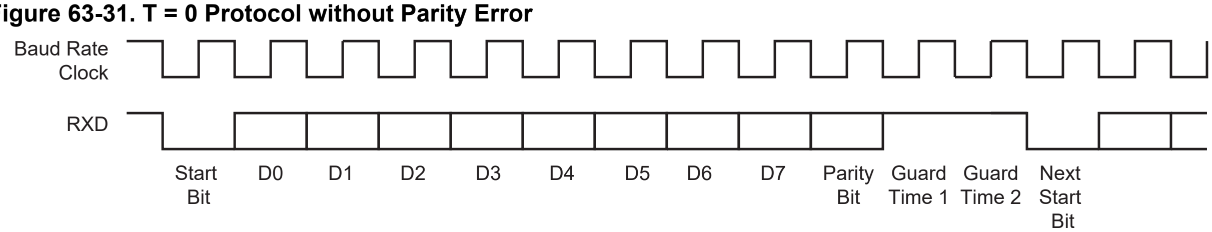
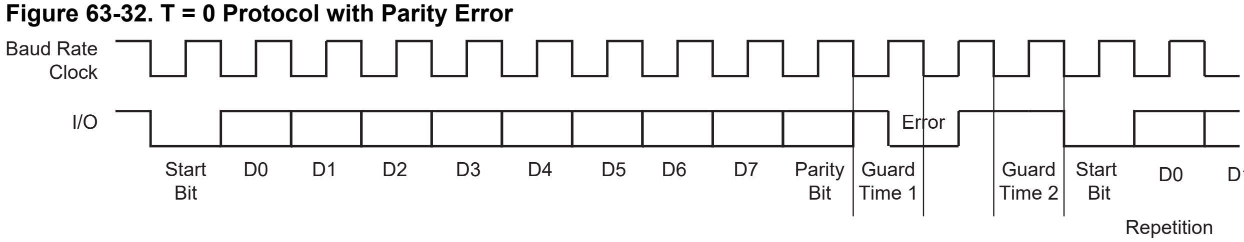


Figure 63-32. T = 0 Protocol with Parity Error



63.7.4.2.1 Receive Error Counter

The USART receiver also records the total number of errors. This can be read in the Number of Error (FLEX_US_NER) register. The NB_ERRORS field can record up to 255 errors. Reading FLEX_US_NER automatically clears the NB_ERRORS field.

63.7.4.2.2 Receive NACK Inhibit

The USART can be configured to inhibit an error. This is done by writing a '1' to FLEX_US_MR.INACK. In this case, no error signal is driven on the I/O line even if a parity bit is detected.

Moreover, if INACK = 1, the erroneous received character is stored in the Receive Holding register as if no error occurred, and the RXRDY bit rises.

63.7.4.2.3 Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next one. Repetition is enabled by writing the FLEX_US_MR.MAX_ITERATION field at a value higher than 0. Each character can be transmitted up to eight times: the first transmission plus seven repetitions.

If MAX_ITERATION does not equal zero, the USART repeats the character as many times as the value loaded in MAX_ITERATION.

When the USART repetition number reaches MAX_ITERATION, and the last repeated character is not acknowledged, the FLEX_US_CSR.ITER bit is set. If the repetition of the character is acknowledged by the receiver, the repetitions are stopped and the iteration counter is cleared.

The FLEX_US_CSR.ITER bit can be cleared by writing the FLEX_US_CR.RSTIT bit to 1.

63.7.4.2.4 Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the FLEX_US_MR.DSNACK bit. The maximum number of NACKs transmitted is programmed in the MAX_ITERATION field. As soon as MAX_ITERATION is reached, no error signal is driven on the I/O line and the FLEX_US_CSR.ITER bit is set.

63.7.4.3 Protocol T = 1

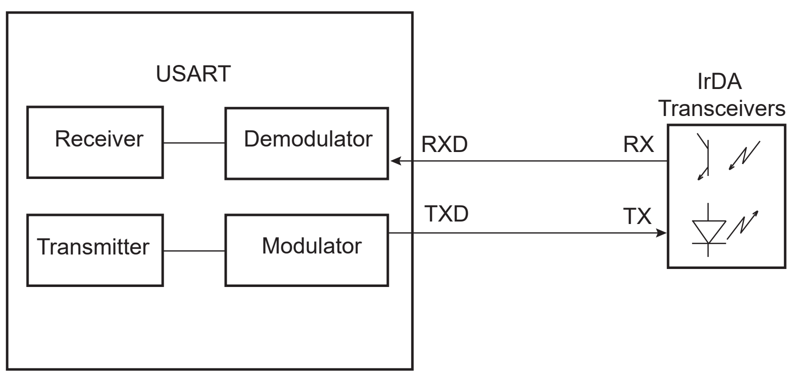
When operating in ISO7816 protocol T = 1, the transmission is similar to an asynchronous format with only one stop bit. The parity is generated when transmitting and checked when receiving. Parity error detection sets the FLEX_US_CSR.PARE bit.

63.7.5 IrDA Mode

The USART features an IrDA mode supplying half-duplex point-to-point wireless communication. It embeds the modulator and demodulator which allows a glueless connection to the infrared transceivers, as shown in the following figure. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kbit/s to 115.2 kbit/s.

The USART IrDA mode is enabled by setting the FLEX_US_MR.USART_MODE field to the value 0x8. The IrDA Filter register (FLEX_US_IF) allows configuring the demodulator filter. The USART transmitter and receiver operate in a normal Asynchronous mode and all parameters are accessible. Note that the modulator and the demodulator are activated.

Figure 63-33. Connection to IrDA Transceivers



The receiver and the transmitter must be enabled or disabled according to the direction of the transmission to be managed.

To receive IrDA signals, the following needs to be done:

- Disable TX and Enable RX
- Configure the TXD pin as PIO and set it as an output to 0 (to avoid LED transmission). Disable the internal pullup (better for power consumption).
- Receive data

63.7.5.1 IrDA Modulation

For baud rates up to and including 115.2 kbit/s, the RZI modulation scheme is used. “0” is represented by a light pulse of 3/16th of a bit time. Some examples of signal pulse duration are shown in the following table.

Table 63-10. IrDA Pulse Duration

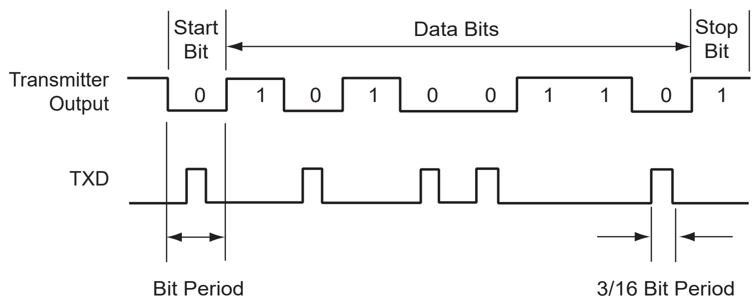
Baud Rate	Pulse Duration (3/16)
2.4 kbit/s	78.13 μs

...........continued

Baud Rate	Pulse Duration (3/16)
9.6 kbit/s	19.53 μs
19.2 kbit/s	9.77 μs
38.4 kbit/s	4.88 μs
57.6 kbit/s	3.26 μs
115.2 kbit/s	1.63 μs

The following figure shows an example of character transmission.

Figure 63-34. IrDA Modulation



63.7.5.2 IrDA Baud Rate

The following table gives some examples of CD values, baud rate error and pulse duration. Note that the requirement on the maximum acceptable error of ±1.87% must be met.

Table 63-11. IrDA Baud Rate Error

Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (μs)
3,686,400	115,200	2	0.00%	1.63
20,000,000	115,200	11	1.38%	1.63
32,768,000	115,200	18	1.25%	1.63
40,000,000	115,200	22	1.38%	1.63
3,686,400	57,600	4	0.00%	3.26
20,000,000	57,600	22	1.38%	3.26
32,768,000	57,600	36	1.25%	3.26
40,000,000	57,600	43	0.93%	3.26
3,686,400	38,400	6	0.00%	4.88
20,000,000	38,400	33	1.38%	4.88
32,768,000	38,400	53	0.63%	4.88
40,000,000	38,400	65	0.16%	4.88
3,686,400	19,200	12	0.00%	9.77
20,000,000	19,200	65	0.16%	9.77
32,768,000	19,200	107	0.31%	9.77
40,000,000	19,200	130	0.16%	9.77

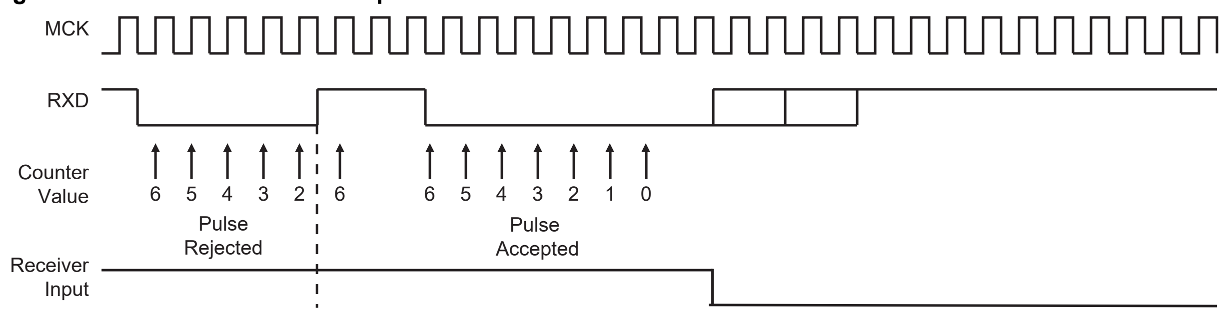
...........continued				
Peripheral Clock	**Baud Rate (bit/s)**	**CD**	**Baud Rate Error**	**Pulse Time (µs)**
3,686,400	9,600	24	0.00%	19.53
20,000,000	9,600	130	0.16%	19.53
32,768,000	9,600	213	0.16%	19.53
40,000,000	9,600	260	0.16%	19.53
3,686,400	2,400	96	0.00%	78.13
20,000,000	2,400	521	0.03%	78.13
32,768,000	2,400	853	0.04%	78.13

63.7.5.3 IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in FLEX_US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the peripheral clock speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with FLEX_US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

The following figure illustrates the operations of the IrDA demodulator.

Figure 63-35. IrDA Demodulator Operations



The programmed value in the FLEX_US_IF register must always meet the following criteria:

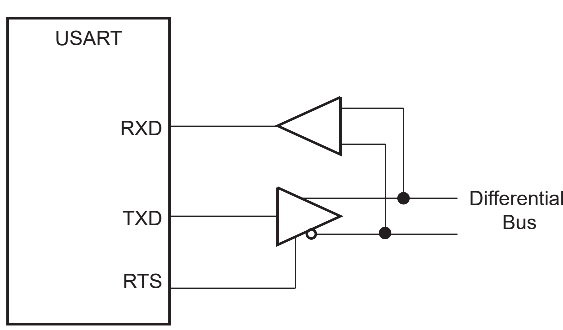
$t_{\text{peripheral clock}} \times (\text{IRDA\_FILTER} + 3) < 1.41\ \mu s$

As the IrDA mode uses the same logic as the ISO7816, note that the FLEX_US_FIDI.FI_DI_RATIO field must be set to a value higher than 0 to make sure IrDA communications operate correctly.

63.7.6 RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in Asynchronous or Synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to an RS485 bus is shown in the following figure.

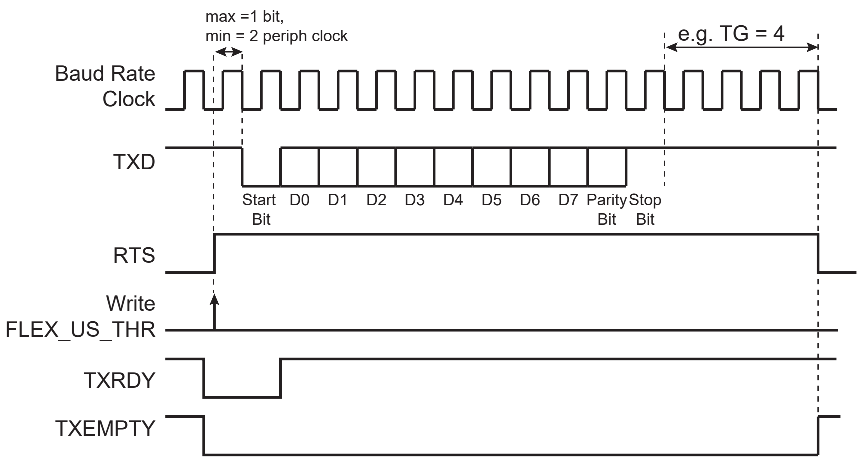
Figure 63-36. Typical Connection to an RS485 Bus



The USART is set to RS485 mode by writing the value 0x1 to the FLEX_US_MR.USART_MODE field.

The RTS pin is at a level inverse to the TXEMPTY bit. Significantly, the RTS pin remains high when a timeguard is programmed, so that the line can remain driven after the last character completion. The following figure gives an example of the RTS waveform during a character transmission when the timeguard is enabled.

Figure 63-37. Example of RTS Drive with Timeguard



Note: In case the minimum period between RTS rising edge and START bit falling edge is not suitable for the application, it is possible to drive the RTS signal by software when FLEX_US_MR.USART_MODE= 0 or 2. FLEX_US_CR.RTSEN/RTSDIS can be configured to manage RTS.

63.7.7 USART Comparison Function on Received Character

The comparison function differs if the asynchronous partial wakeup is enabled or not.

If the asynchronous partial wakeup is disabled, the CMP flag in FLEX_US_CSR is set when the received character matches the conditions programmed in FLEX_US_CMPR. The CMP flag is set as soon as FLEX_US_RHR is loaded with the new received character. The CMP flag is cleared by writing a one to FLEX_US_CR.RSTSTA.

FLEX_US_CMPR can be programmed to provide different comparison methods:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When the FLEX_US_CMPR.CMPMODE bit is set to FLAG_ONLY (value 0), all received data are loaded in FLEX_US_RHR and the CMP flag provides the status of the comparison result.

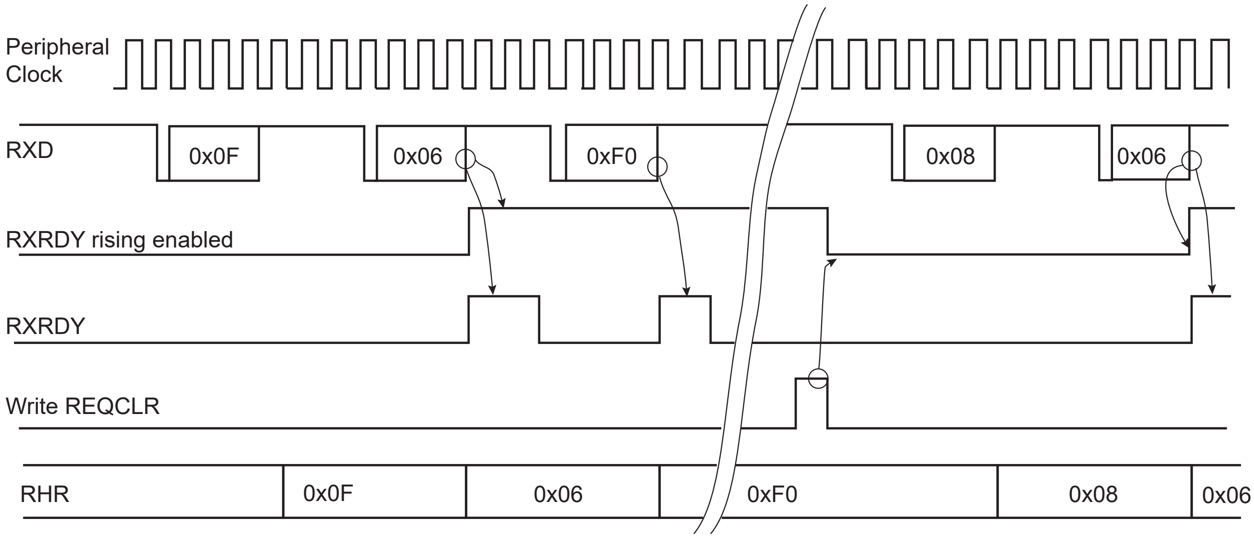
By programming the START_CONDITION.CMPMODE bit (value 1), the comparison function result triggers the start of the loading of FLEX_US_RHR (see the following figure). The trigger condition exists as soon as the received character value matches the condition defined by the programming of VAL1, VAL2 and CMPPAR in FLEX_US_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_US_CR.REQCLR bit.

By setting the CMPMODE bit to FILTER (value 2), the comparison result triggers the start of the US_RHR loading. The trigger condition exists as soon as the received address byte value matches the conditions defined by VAL1, VAL2 in US_CMPR. The comparison trigger event is restarted automatically after the reception of the data byte. This comparison mode is only available when FLEX_US_MR.USART_MODE is set).

The value programmed in the VAL1 and VAL2 fields must not exceed the maximum value of the received character (see CHRL field in FLEX_US_MR).

Figure 63-38. Receive Holding Register Management

CMPMODE = 1, VAL1 = VAL2 = 0x06



63.7.8 USART Asynchronous and Partial Wakeup

Asynchronous and partial wakeup is a means of data preprocessing that qualifies an incoming event, thus allowing the USART to decide whether or not to wake up the system. This operating mode is used primarily when the system is in Wait mode (refer to the section “Power Management Controller (PMC)" for more details) but can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

Asynchronous and partial wakeup requires the USART module to be programmed in UART (FLEX_US_MR.SYNC = 0).

The maximum baud rate that can be achieved when asynchronous and partial wakeup is enabled is 1200.

The FLEX_US_RHR register must be read before enabling the asynchronous and partial wakeup.

When asynchronous and partial wakeup is enabled for the USART (refer to the section “Power Management Controller (PMC)"), the PMC decodes a clock request from the USART. The request is generated as soon as there is a falling edge on the RXD line as this may indicate the beginning of a start bit. If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the USART.

As soon as the clock is provided by the PMC, the USART processes the received frame and compares the received character with VAL1 and VAL2 in FLEX_US_CMPR.

The USART instructs the PMC to disable the clock if the received character value does not meet the conditions defined by VAL1 and VAL2 fields in FLEX_US_CMPR (see figure Asynchronous Event Generating Only Partial Wakeup).

If the received character value meets the conditions, the USART instructs the PMC to exit the system from Wait mode (see figure Asynchronous Wakeup Use Case Examples).

The VAL1 and VAL2 fields can be programmed to provide different comparison methods and thus matching conditions.

- If VAL1 equals VAL2, then the comparison is performed on a single value and the wakeup is triggered if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 wakes up the system.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.
- If VAL1 = 0 and VAL2 = 511, the wakeup is triggered as soon as a character is received.

The matching condition can be configured to include the parity bit (FLEX_US_CMPR.CMPPAR). Thus, if the received data matches the comparison condition defined by VAL1 and VAL2 but a parity error is encountered, the matching condition is cancelled and the UART instructs the PMC to disable the clock (see figure Asynchronous Event Generating Only Partial Wakeup).

If the processor and peripherals are running, the USART can be configured in Asynchronous and Partial Wakeup mode by enabling the PMC_SLPWK_ER (refer to the section “Power Management Controller (PMC)"). When activity is detected on the receive line, the USART requests the clock from the PMC and the comparison is performed. If there is a comparison match, the USART continues to request the clock. If there is no match, the clock is switched off for the USART only, until a new activity is detected.

The CMPMODE configuration has no effect when Asynchronous and Partial Wakeup mode is enabled for the USART (refer to PMC_SLPWK_ER in the section “Power Management Controller (PMC)").

When the system is in Active mode and the USART enters Asynchronous and Partial Wakeup mode, the flag RXRDY must be programmed as the unique source of the USART interrupt.

When the system exits Wait mode as the result of a matching condition, the RXRDY flag is used to determine if the USART is the source for the exit from Wait mode.

Figure 63-39. Asynchronous Wakeup Use Case Examples

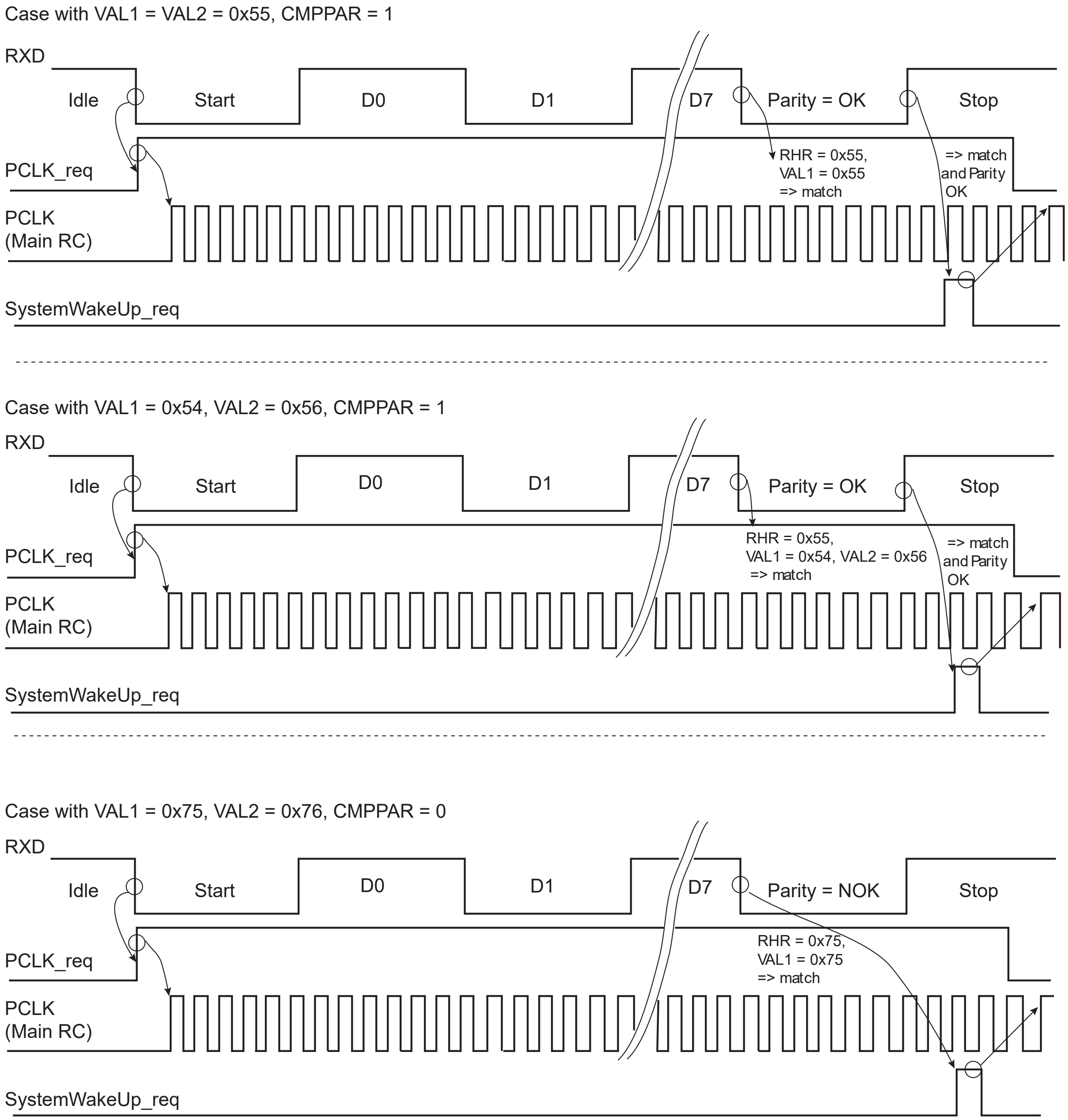
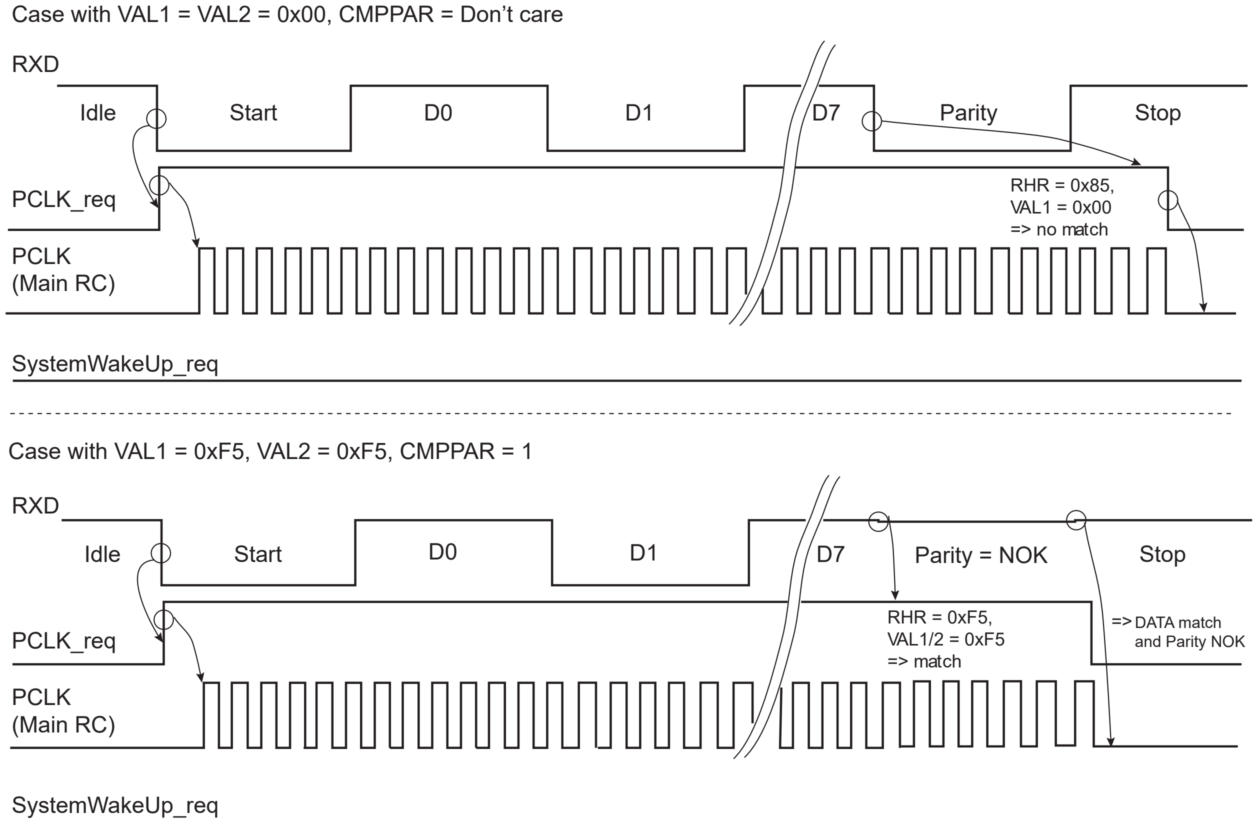


Figure 63-40. Asynchronous Event Generating Only Partial Wakeup



63.7.9 LIN Mode

The LIN mode provides host node and client node connectivity on a LIN bus.

The LIN (Local Interconnect Network) is a serial communication protocol which efficiently supports the control of mechatronic nodes in distributed automotive applications.

The main properties of the LIN bus are:

- Single host/multiple clients concept
- Low-cost silicon implementation based on common UART/SCI interface hardware, an equivalent in software, or as a pure state machine.
- Self synchronization without quartz or ceramic resonator in the client nodes
- Deterministic signal transmission
- Low cost single-wire implementation
- Speed up to 20 kbit/s

LIN provides cost efficient bus communication where the bandwidth and versatility of CAN are not required.

The LIN mode enables processing LIN frames with a minimum of action from the microprocessor.

63.7.9.1 Modes of Operation

The USART can act either as a LIN host node or as a LIN client node.

The node configuration is chosen by setting the USART_MODE field in the USART Mode register (FLEX_US_MR):

- LIN host node (USART_MODE = 0xA)
- LIN client node (USART_MODE = 0xB)

In order to avoid unpredictable behavior, any change of the LIN node configuration must be followed by a software reset of the transmitter and of the receiver (except the initial node configuration after a hardware reset). See Receiver and Transmitter Control.

63.7.9.2 Baud Rate Configuration

See Baud Rate in Asynchronous Mode.

- LIN host node: The baud rate is configured in FLEX_US_BRGR.
- LIN client node: The initial baud rate is configured in FLEX_US_BRGR. This configuration is automatically copied in the LIN Baud Rate register (FLEX_US_LINBRR) when writing FLEX_US_BRGR. After the synchronization procedure, the baud rate is updated in FLEX_US_LINBRR.

63.7.9.3 Receiver and Transmitter Control

See Receiver and Transmitter Control.

63.7.9.4 Character Transmission

See Transmitter Operations.

63.7.9.5 Character Reception

See Receiver Operations.

63.7.9.6 Header Transmission (Host Node Configuration)

All LIN frames start with a header sent by the host node and consisting of a Synch Break Field, a Synch Field and an Identifier Field.

So in host node configuration, the frame handling starts with the sending of the header.

The header is transmitted as soon as the identifier is written in the LIN Identifier register (FLEX_US_LINIR). At this moment, the flag TXRDY falls.

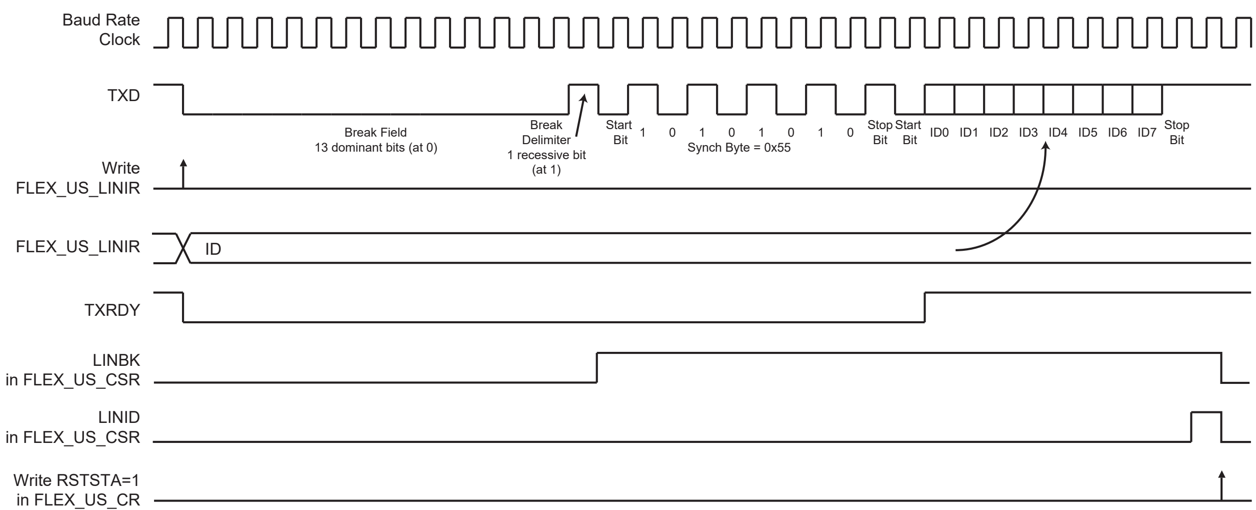
The Break Field, the Synch Field and the Identifier Field are sent automatically one after the other.

The Break Field consists of 13 dominant bits and 1 recessive bit, the Synch Field is the character 0x55 and the Identifier corresponds to the character written in the LIN Identifier register (FLEX_US_LINIR). The Identifier parity bits can be automatically computed and sent (see Identifier Parity).

The flag TXRDY rises when the identifier character is transferred into the shift register of the transmitter.

As soon as the Synch Break Field is transmitted, the FLEX_US_CSR.LINBK flag bit is set. Likewise, as soon as the Identifier Field is sent, the FLEX_US_CSR.LINID flag bit is set. These flags are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

Figure 63-41. Header Transmission



63.7.9.7 Header Reception (Client Node Configuration)

All the LIN frames start with a header which is sent by the host node and consists of a Synch Break Field, Synch Field and Identifier Field.

In client node configuration, the frame handling starts with the reception of the header.

The USART uses a break detection threshold of 11 nominal bit times at the actual baud rate. At any time, if 11 consecutive recessive bits are detected on the bus, the USART detects a Break Field. As long as a Break Field has not been detected, the USART stays idle and the received data are not taken in account.

When a Break Field has been detected, the FLEX_US_CSR.LINBK flag is set and the USART expects the Synch Field character to be 0x55. This field is used to update the actual baud rate in order to remain synchronized (see Client Node Synchronization). If the received Synch character is not 0x55, an Inconsistent Synch Field error is generated (see LIN Errors).

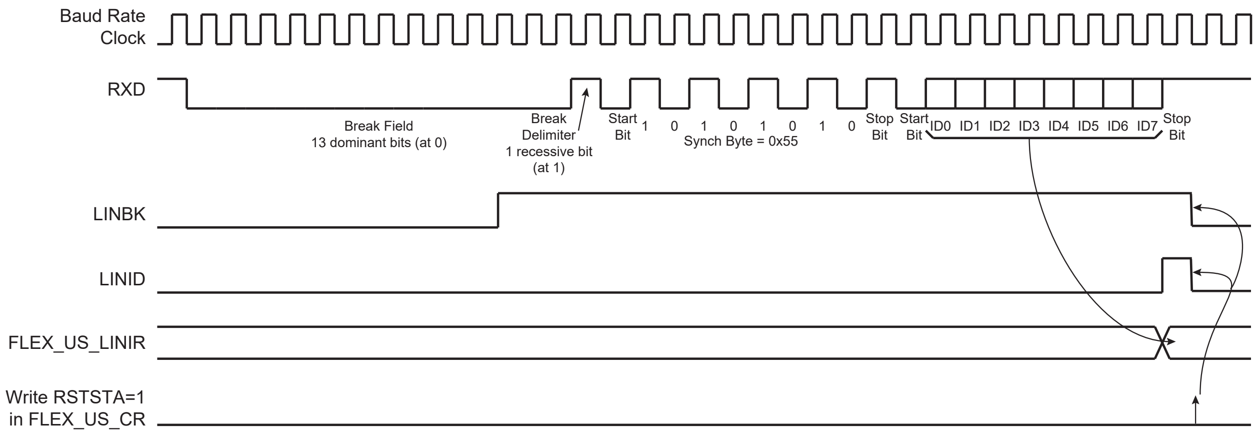
After receiving the Synch Field, the USART expects to receive the Identifier Field.

When the Identifier Field has been received, the FLEX_US_CSR.LINID flag bit is set. At this moment, the IDCHR field in the LIN Identifier register (FLEX_US_LINIR) is updated with the received character. The Identifier parity bits can be automatically computed and checked (see Identifier Parity).

If the header is not entirely received within the time given by the maximum length of the header $t_{Header\_Maximum}$, the FLEX_US_CSR.LINHTE error flag bit is set.

The flag bits LINID, LINBK and LINHTE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

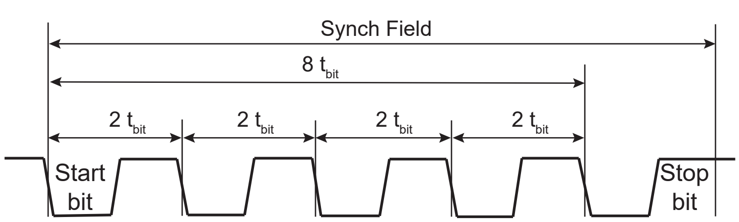
Figure 63-42. Header Reception



63.7.9.8 Client Node Synchronization

The synchronization is done only in client node configuration. The procedure is based on time measurement between the falling edges of the Synch Field. The falling edges are available in distances of 2, 4, 6 and 8 bit times.

Figure 63-43. Synch Field



The time measurement is made by a 19-bit counter driven by the sampling clock (see Baud Rate Generator).

When the start bit of the Synch Field is detected, the counter is reset. Then during the next eight t_{bit} of the Synch Field, the counter is incremented. At the end of these eight t_{bit}, the counter is stopped. At this moment, the 16 most significant bits of the counter (value divided by 8) give the new clock divider (LINCD) and the 3 least significant bits of this value (the remainder) give the new fractional part (LINFP).

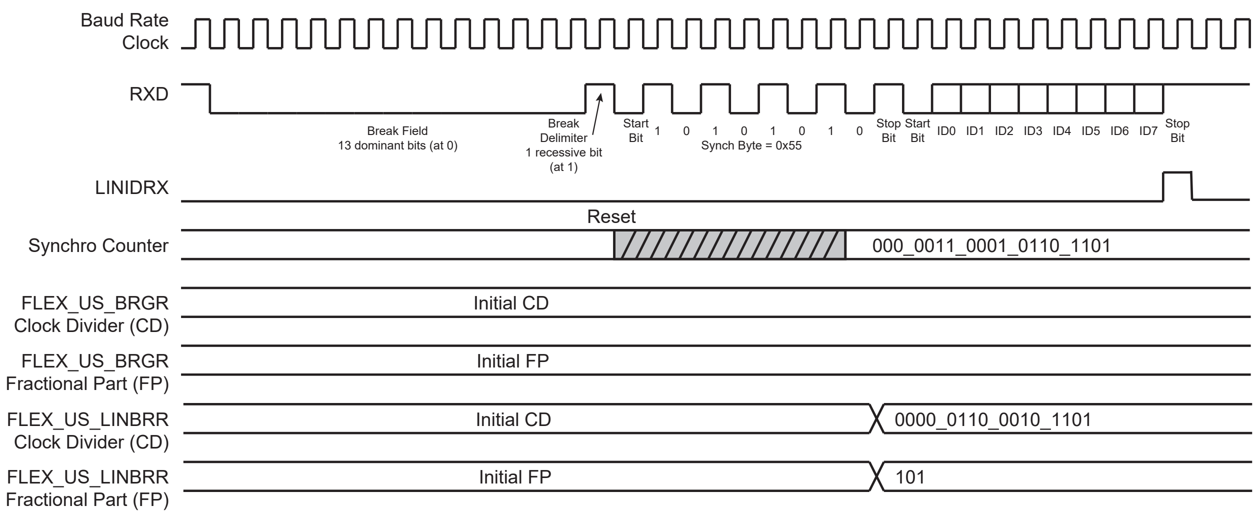
Once the Synch Field has been entirely received, the clock divider (LINCD) and the fractional part (LINFP) are updated in the LIN Baud Rate register (FLEX_US_LINBRR) with the computed values, if the Synchronization is not disabled by the SYNCDIS bit in the LIN Mode register (FLEX_US_LINMR).

After reception of the Synch Field:

- If it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%), then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINSTE error flag bit is set.
- If it appears that the sampled Synch character is not equal to 0x55, then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINISFE error flag bit is set.

Flags LINSTE and LINISFE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

Figure 63-44. Client Node Synchronization



The synchronization accuracy depends on several parameters:

- The nominal clock frequency (f_{Nom}) (the theoretical client node clock frequency)
- The baud rate
- The oversampling (OVER = 0 => 16X or OVER = 1 => 8X)

The following formula is used to compute the deviation of the client bit rate relative to the host bit rate after synchronization (f_{CLIENT} is the real client node clock frequency).

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times \text{f}_{\text{CLIENT}}}\right)\%$$

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times \left(\frac{\text{f}_{\text{TOL\_UNSYNCH}}}{100}\right) \times \text{f}_{\text{Nom}}}\right)\%$$

$$-0.5 \le \alpha \le +0.5 \quad -1 < \beta < +1$$

$f_{TOL\_UNSYNCH}$ is the deviation of the real client node clock from the nominal clock frequency. The LIN Standard imposes that it must not exceed ±15%. The LIN Standard imposes also that for communication between two nodes, their bit rate must not differ by more than ±2%. This means that the baud rate deviation must not exceed ±1%.

Therefore, a minimum value for the nominal clock frequency can be computed as follows:

$$\text{f}_{\text{Nom}}(\text{min}) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{Over}) + 1] \times \text{Baud rate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

Examples:

- Baud rate = 20 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 2.64 MHz
- Baud rate = 20 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 1.47 MHz
- Baud rate = 1 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 132 kHz
- Baud rate = 1 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 74 kHz

63.7.9.9 Identifier Parity

A protected identifier consists of two subfields: the identifier and the identifier parity. Bits 0 to 5 are assigned to the identifier, and bits 6 and 7 are assigned to the parity.

The USART interface can generate/check these parity bits, but this feature can also be disabled. The user can choose between two modes via the FLEX_US_LINMR.PARDIS bit:

- PARDIS = 0:
  – During header transmission, the parity bits are computed and sent with the six least significant bits of the IDCHR field of the LIN Identifier register (FLEX_US_LINIR). Bits 6 and 7 of this register are discarded.
  – During header reception, the parity bits of the identifier are checked. If the parity bits are wrong, an Identifier Parity error occurs (see Parity). Only the six least significant bits of the IDCHR field are updated with the received Identifier. Bits 6 and 7 are stuck to 0.
- PARDIS = 1:
  – During header transmission, all the bits of the IDCHR field of the LIN Identifier register (FLEX_US_LINIR) are sent on the bus.
  – During header reception, all the bits of the IDCHR field are updated with the received Identifier.

63.7.9.10 Node Action

Depending on the identifier, the node is affected—or not—by the LIN response. Consequently, after sending or receiving the identifier, the USART must be configured. There are three possible configurations:

- PUBLISH: the node sends the response.
- SUBSCRIBE: the node receives the response.
- IGNORE: the node is not concerned by the response, it does not send and does not receive the response.

This configuration is made by the LIN Node Action (NACT) field in USART LIN Mode Register (FLEX_US_LINMR).

Example: a LIN cluster that contains a host and two clients:

- Data transfer from the host to client 1 and to client 2:

NACT(host) = PUBLISH

NACT(client 1) = SUBSCRIBE

NACT(client 2) = SUBSCRIBE

- Data transfer from the host to client 1 only:

NACT(host) = PUBLISH

NACT(client 1) = SUBSCRIBE

NACT(client 2) = IGNORE

- Data transfer from client 1 to the host:

NACT(host) = SUBSCRIBE

NACT(client 1) = PUBLISH

NACT(client 2) = IGNORE

- Data transfer from client 1 to client 2:

NACT(host) = IGNORE

NACT(client 1) = PUBLISH

NACT(client 2) = SUBSCRIBE

- Data transfer from client 2 to the host and to client 1:

NACT(host) = SUBSCRIBE

NACT(client 1) = SUBSCRIBE

NACT(client 2) = PUBLISH

63.7.9.11 Response Data Length

The LIN response data length is the number of data fields (bytes) of the response excluding the checksum.

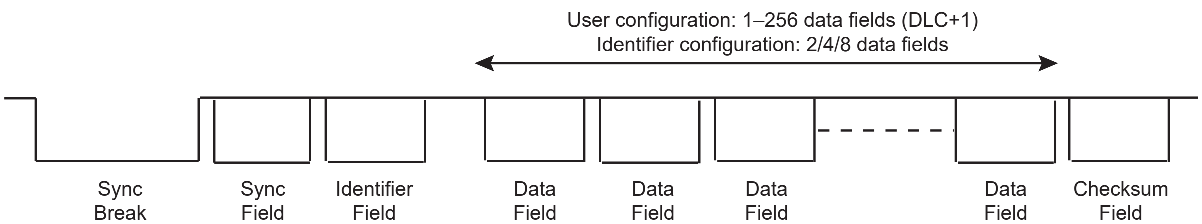
The response data length can either be configured by the user or be defined automatically by bits 4 and 5 of the Identifier (compatibility to LIN Specification 1.1). The user can choose between these two modes by the FLEX_US_LINMR.DLM bit:

- DLM = 0: The response data length is configured by the user via the FLEX_US_LINMR.DLC field. The response data length is equal to (DLC + 1) bytes. DLC can be programmed from 0 to 255, so the response can contain from 1 data byte up to 256 data bytes.
- DLM = 1: The response data length is defined by the Identifier (IDCHR in FLEX_US_LINIR) according to the table below. The FLEX_US_LINMR.DLC field is discarded. The response can contain 2 or 4 or 8 data bytes.

Table 63-12. Response Data Length if DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length (bytes)
0	0	2
0	1	2
1	0	4
1	1	8

Figure 63-45. Response Data Length



63.7.9.12 Checksum

The last field of a frame is the checksum. The checksum contains the inverted 8- bit sum with carry, over all data bytes or all data bytes and the protected identifier. Checksum calculation over the data bytes only is called classic checksum and it is used for communication with LIN 1.3 clients. Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for communication with LIN 2.0 clients.

The USART can be configured to:

- Send/Check an Enhanced checksum automatically (CHKDIS = 0 & CHKTYP = 0)
- Send/Check a Classic checksum automatically (CHKDIS = 0 & CHKTYP = 1)
- Not send/check a checksum (CHKDIS = 1)

This configuration is made by the Checksum Type (CHKTYP) and Checksum Disable (CHKDIS) bits of FLEX_US_LINMR.

If the checksum feature is disabled, the user can send it manually all the same, by considering the checksum as a normal data byte and by adding 1 to the response data length (see Response Data Length).

63.7.9.13 Frame Slot Mode

This mode is useful only for host nodes. It respects the following rule: each frame slot shall be longer than or equal to $t_{Frame\_Maximum}$.

If the Frame Slot mode is enabled (FSDIS = 0) and a frame transfer has been completed, the TXRDY flag is set again only after $t_{Frame\_Maximum}$ delay, from the start of frame. So the host node cannot send a new header if the frame slot duration of the previous frame is inferior to $t_{Frame\_Maximum}$.

If the Frame Slot mode is disabled (FSDIS = 1) and a frame transfer has been completed, the TXRDY flag is set again immediately.

The $t_{Frame\_Maximum}$ is calculated as follows:

If the Checksum is sent (CHKDIS = 0):

- $t_{Header\_Nominal} = 34 \times t_{bit}$
- $t_{Response\_Nominal} = 10 \times (NData + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$[(1)]
- $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1 + 1) + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = (77 + 14 \times DLC) \times t_{bit}$

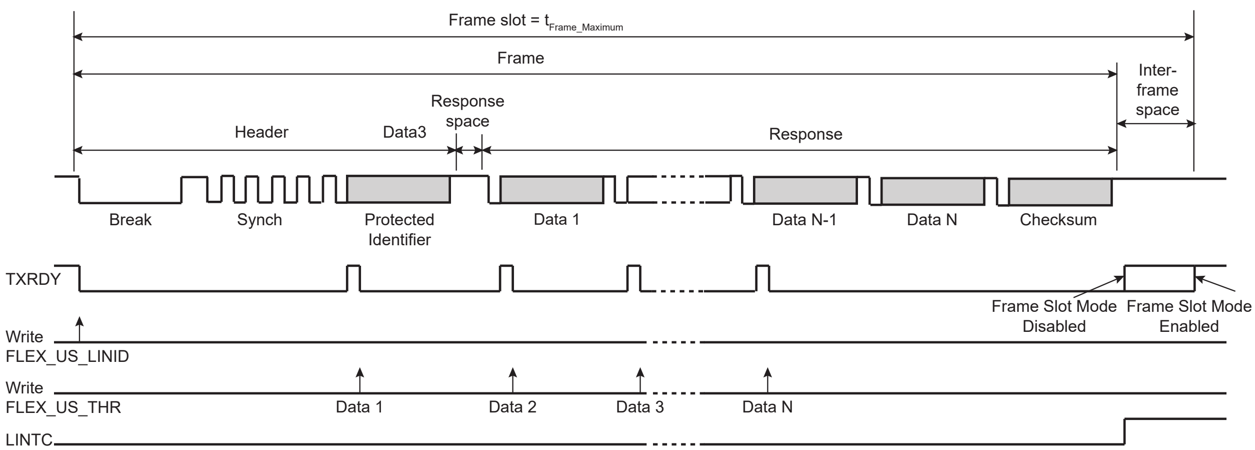
If the Checksum is not sent (CHKDIS = 1):

- $t_{Header\_Nominal} = 34 \times t_{bit}$
- $t_{Response\_Nominal} = 10 \times NData \times t_{bit}$
- $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$[(1)]
- $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1) + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = (63 + 14 \times DLC) \times t_{bit}$

Note:

1. The term “+1” leads to an integer result for $t_{Frame\_Maximum}$ (LIN Specification 1.3).

Figure 63-46. Frame Slot Mode



63.7.9.14 LIN Errors

63.7.9.14.1 Bit Error

This error is generated in host or client node configuration, when the USART is transmitting and if the transmitted value on the Tx line is different from the value sampled on the Rx line. If a bit error is detected, the transmission is aborted at the next byte border.

This error is reported by the FLEX_US_CSR.LINBE flag.

63.7.9.14.2 Inconsistent Synch Field Error

This error is generated in client node configuration, if the Synch Field character received is other than 0x55.

This error is reported by the FLEX_US_CSR.LINISFE flag.

63.7.9.14.3 Identifier Parity Error

This error is generated in client node configuration, if the parity of the identifier is wrong. This error can be generated only if the parity feature is enabled (PARDIS = 0).

This error is reported by the FLEX_US_CSR.LINIPE flag.

63.7.9.14.4 Checksum Error

This error is generated in host or client node configuration, if the received checksum is wrong. This flag can be set to 1 only if the checksum feature is enabled (CHKDIS = 0).

This error is reported by the FLEX_US_CSR.LINCE flag.

63.7.9.14.5 Client Not Responding Error

This error is generated in host or client node configuration, when the USART expects a response from another node (NACT = SUBSCRIBE) but no valid message appears on the bus within the time given by the maximum length of the message frame, $t_{Frame\_Maximum}$ (see Frame Slot Mode). This error is disabled if the USART does not expect any message (NACT = PUBLISH or NACT = IGNORE).

This error is reported by the FLEX_US_CSR.LINSNRE.

63.7.9.14.6 Synch Tolerance Error

This error is generated in client node configuration if, after the clock synchronization procedure, it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%).

This error is reported by the FLEX_US_CSR.LINSTE flag.

63.7.9.14.7 Header Timeout Error

This error is generated in client node configuration, if the header is not entirely received within the time given by the maximum length of the header, $t_{Header\_Maximum}$.

This error is reported by the FLEX_US_CSR.LINHTE flag.

63.7.9.15 LIN Frame Handling

63.7.9.15.1 Host Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the host node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS and DLC in FLEX_US_LINMR to configure the frame transfer.
- Check that FLEX_US_CSR. TXRDY is set to 1.
- Write FLEX_US_LINIR.IDCHR to send the header.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the USART sends the response.
  – Wait until FLEX_US_CSR. TXRDY rises.
  – Write FLEX_US_THR.TCHR to send a byte.
  – If all the data have not been written, repeat the two previous steps.
  – Wait until FLEX_US_CSR.LINTC rises.
  – Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  – Wait until FLEX_US_CSR.RXRDY rises.
  – Read FLEX_US_RHR.RCHR.
  – If all the data have not been read, repeat the two previous steps.
  – Wait until FLEX_US_CSR.LINTC rises.
  – Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  – Wait until FLEX_US_CSR.LINTC rises.
  – Check the LIN errors.

Figure 63-47. Host Node Configuration, NACT = PUBLISH

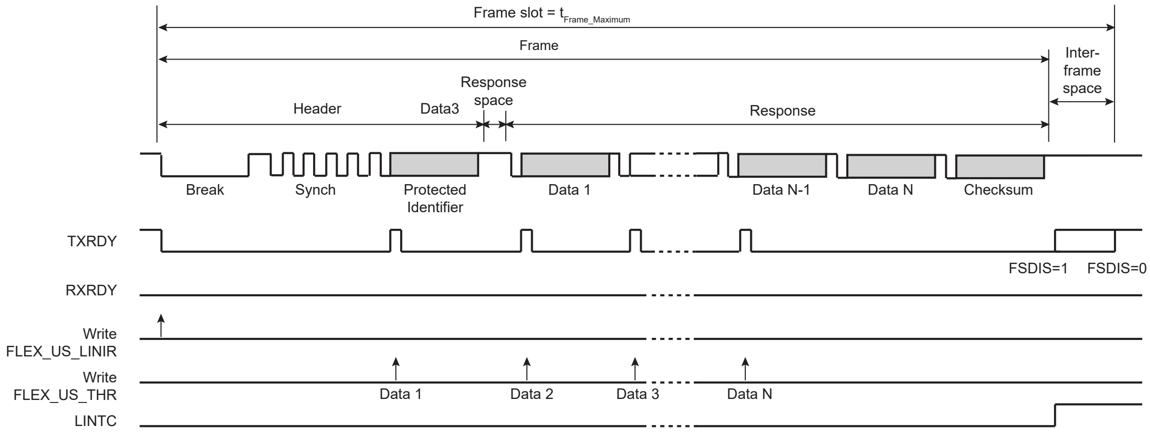


Figure 63-48. Host Node Configuration, NACT = SUBSCRIBE

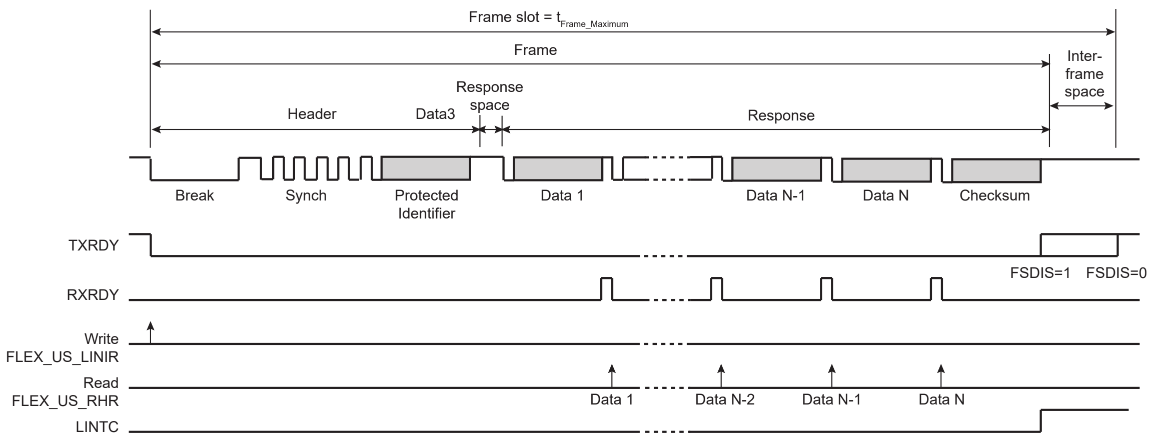
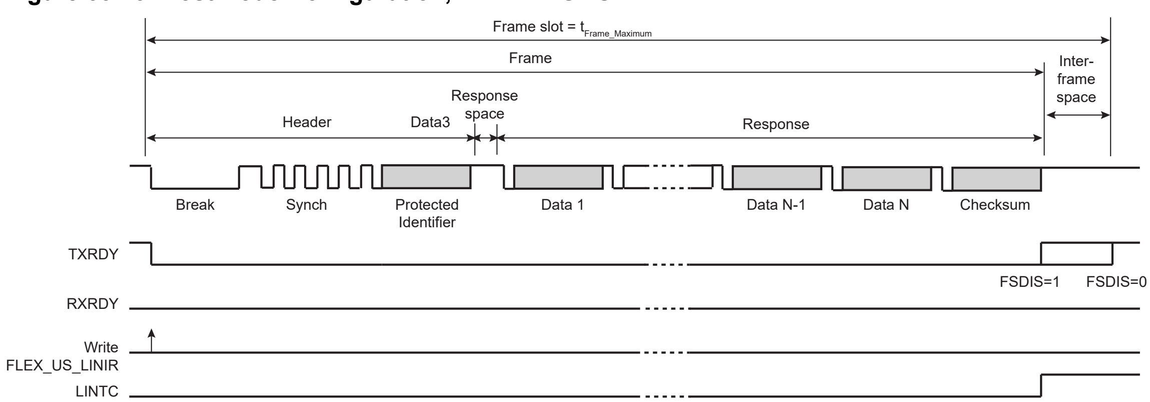


Figure 63-49. Host Node Configuration, NACT = IGNORE



63.7.9.15.2 Client Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the client node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Wait until FLEX_US_CSR.LINID rises.
- Check LINISFE and LINPE errors.
- Read FLEX_US_RHR.IDCHR.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC in FLEX_US_LINMR to configure the frame transfer.

IMPORTANT: If the NACT configuration for this frame is PUBLISH, FLEX_US_LINMR must be written with NACT = PUBLISH even if this field is already correctly configured, in order to set the TXREADY flag and the corresponding write transfer request.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the LIN controller sends the response.
  – Wait until FLEX_US_CSR.TXRDY rises.
  – Write FLEX_US_THR.TCHR to send a byte.
  – If all the data have not been written, repeat the two previous steps.
  – Wait until FLEX_US_CSR. LINTC rises.
  – Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  – Wait until FLEX_US_CSR.RXRDY rises.
  – Read FLEX_US_RHR.RCHR.
  – If all the data have not been read, repeat the two previous steps.
  – Wait until FLEX_US_CSR.LINTC rises.
  – Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  – Wait until FLEX_US_CSR.LINTC rises.
  – Check the LIN errors.

Figure 63-50. Client Node Configuration, NACT = PUBLISH

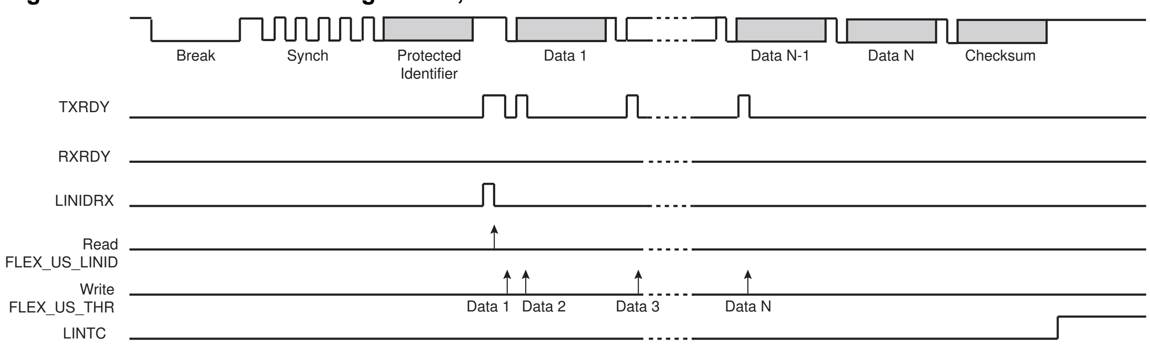


Figure 63-51. Client Node Configuration, NACT = SUBSCRIBE

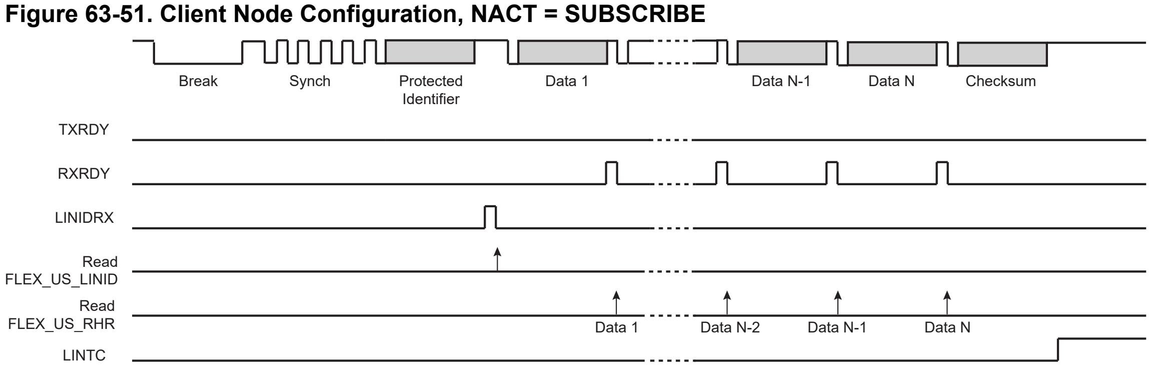
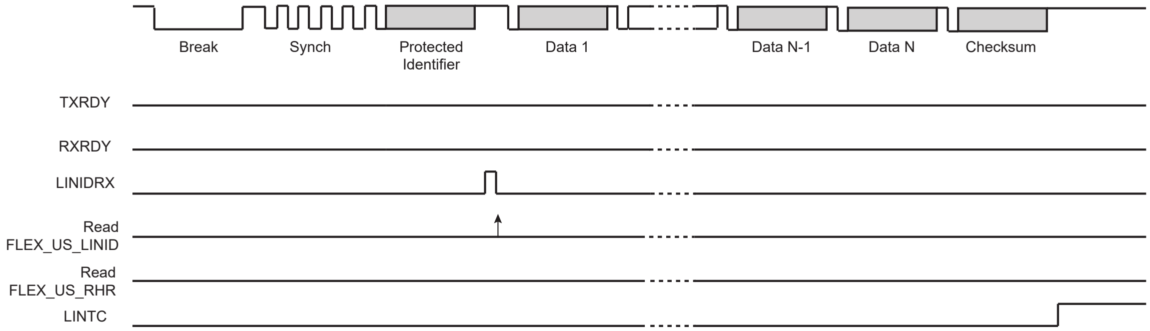


Figure 63-52. Client Node Configuration, NACT = IGNORE



63.7.9.16 LIN Frame Handling with the DMAC

The USART can be used in association with the DMAC in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The DMAC uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The DMAC always writes in the Transmit Holding register (FLEX_US_THR) and it always reads in the Receive Holding register (FLEX_US_RHR). The size of the data written or read by the DMAC in the USART is always a byte.

63.7.9.16.1 Host Node Configuration

The user can choose between two DMAC modes by configuring the FLEX_US_LINMR.PDCM bit:

- PDCM = 1: The LIN configuration is stored in the WRITE buffer and it is written by the DMAC in the Transmit Holding register FLEX_US_THR (instead of the LIN Mode register FLEX_US_LINMR). Because the DMAC transfer size is limited to a byte, the transfer is split into two accesses. During the first access, the NACT, PARDIS, CHKDIS, CHKTYP, DLM and FSDIS bits are written. During the second access, the 8-bit DLC field is written.
- PDCM = 0: The LIN configuration is not stored in the WRITE buffer and it must be written by the user in FLEX_US_LINMR.

The WRITE buffer also contains the Identifier and the data, if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

Figure 63-53. Host Node with DMAC (PDCM = 1)

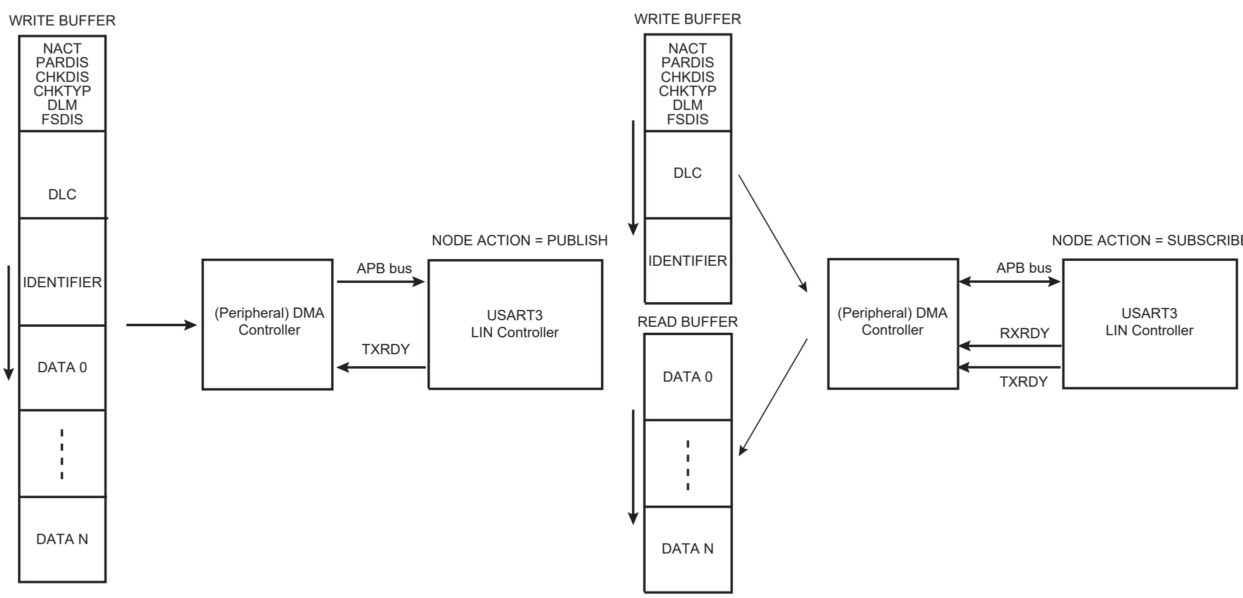
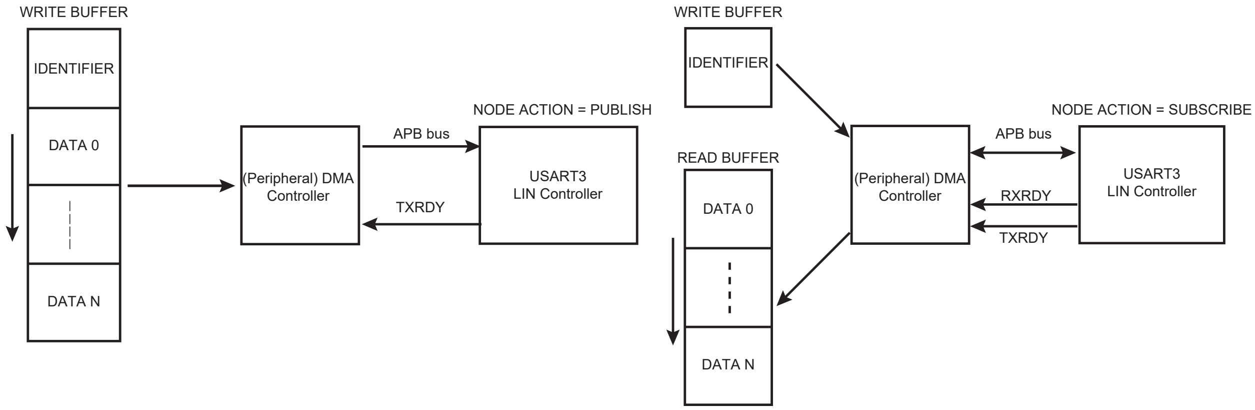


Figure 63-54. Host Node with DMAC (PDCM = 0)



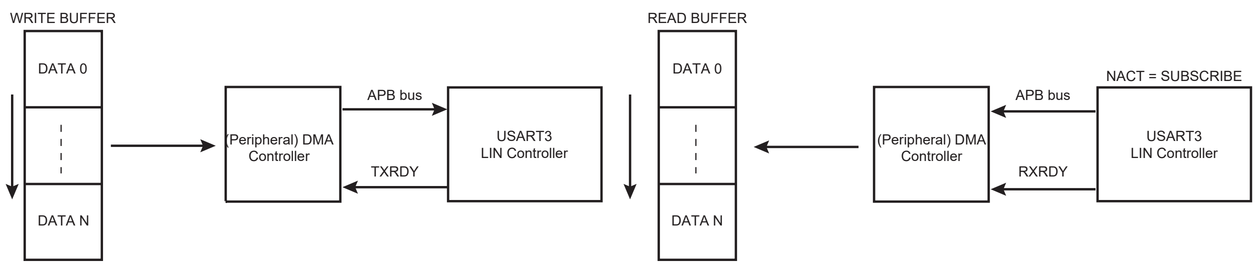
63.7.9.16.2 Client Node Configuration

In this configuration, the DMAC transfers only the data. The identifier must be read by the user in the LIN Identifier register (FLEX_US_LINIR). The LIN mode must be written by the user in FLEX_US_LINMR.

The WRITE buffer contains the data if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

Figure 63-55. Client Node with DMAC



63.7.9.17 Wakeup Request

Any node in a sleeping LIN cluster may request a wakeup.

In the LIN 2.0 specification, the wakeup request is issued by forcing the bus to the dominant state from 250 μs to 5 ms. For this, it is necessary to send the character 0xF0 in order to impose five successive dominant bits. Whatever the baud rate is, this character respects the specified timings.

- Baud rate min = 1 kbit/s -> t_{bit} = 1 ms -> 5 t_{bit} = 5 ms
- Baud rate max = 20 kbit/s -> t_{bit} = 50 μs -> 5 t_{bit} = 250 μs

In the LIN 1.3 specification, the wakeup request should be generated with the character 0x80 in order to impose eight successive dominant bits.

Using the FLEX_US_LINMR.WKUPTYP bit, the user can choose to send either a LIN 2.0 wakeup request (WKUPTYP = 0) or a LIN 1.3 wakeup request (WKUPTYP = 1).

A wakeup request is transmitted by writing the FLEX_US_CR.LINWKUP bit to 1. Once the transfer is completed, the LINTC flag is asserted in the Status register (FLEX_US_CSR). It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

63.7.9.18 Bus Idle Timeout

If the LIN bus is inactive for a certain duration, the client nodes shall automatically enter in Sleep mode. In the LIN 2.0 specification, this timeout is defined as 4 seconds. In the LIN 1.3 specification, it is defined as 25,000 t_{bit}.

In client node configuration, the receiver timeout detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver to go into Sleep mode.

The timeout delay period (during which the receiver waits for a new character) is programmed in the FLEX_US_RTOR.TO field. If a zero is written to the TO field, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 17-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises.

If STTTO is performed, the counter clock is stopped until a first character is received.

If RETTO is performed, the counter starts counting down immediately from the value TO.

Table 63-13. Receiver Timeout Programming

LIN Specification	Baud Rate	Timeout period	TO
2.0	1,000 bit/s	4s	4,000
	2,400 bit/s		9,600
	9,600 bit/s		38,400
	19,200 bit/s		76,800
	20,000 bit/s		80,000
1.3	–	25,000 t_{bit}	25,000

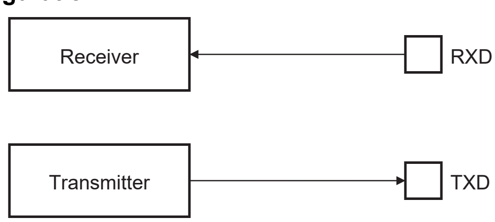
63.7.10 Test Modes

The USART can be programmed to operate in three different test modes. The internal loopback capability allows on-board diagnostics. In Loopback mode, the USART interface pins are disconnected or not and reconfigured for loopback internally or externally.

63.7.10.1 Normal Mode

Normal mode connects the RXD pin on the receiver input and the transmitter output on the TXD pin.

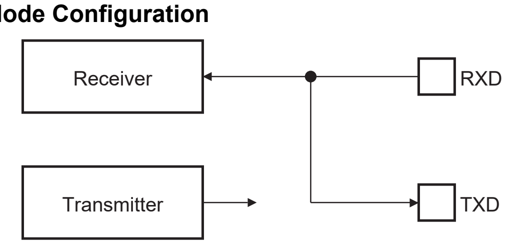
Figure 63-56. Normal Mode Configuration



63.7.10.2 Automatic Echo Mode

Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the RXD pin, it is sent to the TXD pin, as shown in the following figure. Programming the transmitter has no effect on the TXD pin. The RXD pin is still connected to the receiver input, thus the receiver remains active.

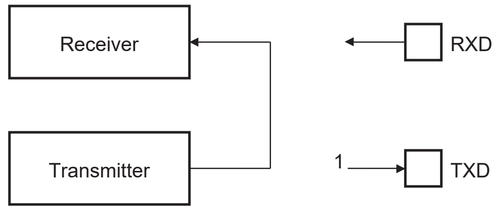
Figure 63-57. Automatic Echo Mode Configuration



63.7.10.3 Local Loopback Mode

Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the following figure. The TXD and RXD pins are not used. The RXD pin has no effect on the receiver and the TXD pin is continuously driven high, as in idle state.

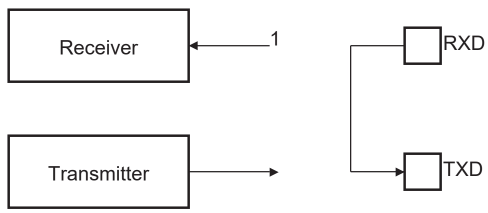
Figure 63-58. Local Loopback Mode Configuration



63.7.10.4 Remote Loopback Mode

Remote Loopback mode directly connects the RXD pin to the TXD pin, as shown in the following figure. The transmitter and the receiver are disabled and have no effect. This mode allows bit-by-bit retransmission.

Figure 63-59. Remote Loopback Mode Configuration



63.7.11 USART FIFOs

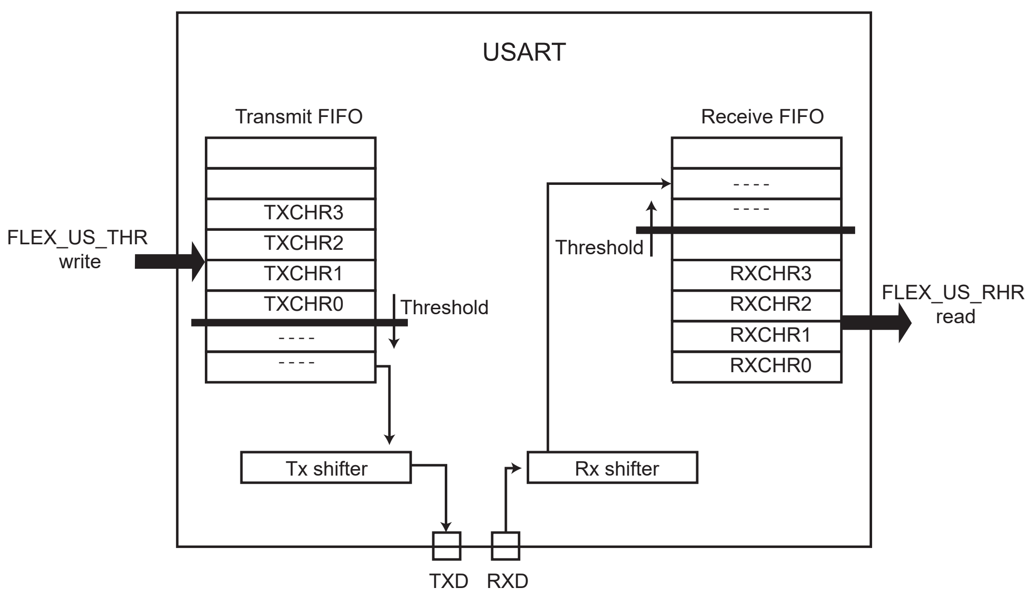
63.7.11.1 Overview

The USART includes two FIFOs which can be enabled/disabled using FLEX_US_CR.FIFOEN/FIFODIS. Both the transmitter and the receiver must be disabled before enabling or disabling the FIFOs, using the FLEX_US_CR.TXDIS/RXDIS bits.

Writing FLEX_US_CR.FIFOEN to ‘1’ enables a 32-data Transmit FIFO and a 32-data Receive FIFO.

When the FIFO is enabled, it is possible to write or to read single data (5-bit to 9-bit data) or multiple data (5-bit to 8-bit data) in the same access to FLEX_US_THR/RHR. See FIFO Single Data Access and FIFO Multiple Data Access.

Figure 63-60. USART FIFOs Block Diagram



63.7.11.2 Sending Data with FIFO Enabled

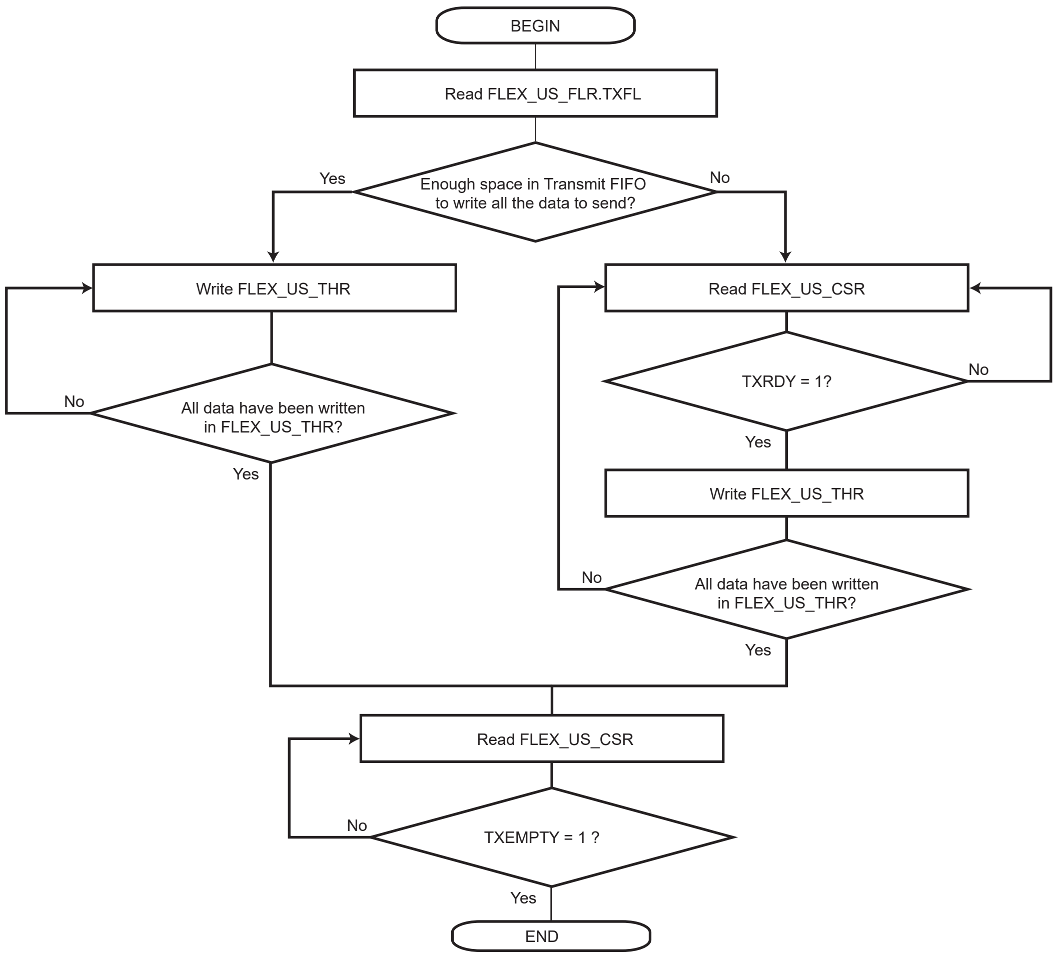
When the Transmit FIFO is enabled, write access to FLEX_US_THR loads the Transmit FIFO.

The FIFO level is provided in FLEX_US_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_US_CSR.TXRDY and the data can be successively written in FLEX_US_THR.

If the FIFO cannot accept the data due to insufficient space, wait for the TXRDY flag to be set before writing the data in FLEX_US_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

Figure 63-61. Sending Data with FIFO Enabled

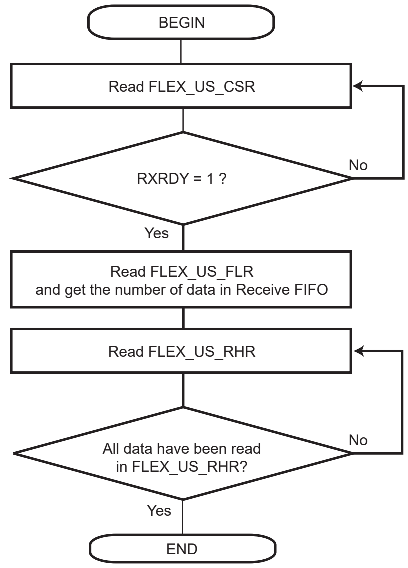


63.7.11.3 Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_US_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to ‘1’), the exact number of data can be checked with FLEX_US_FLR.RXFL. All the data can be read successively in FLEX_US_RHR without checking the RXRDY flag between each access.

Figure 63-62. Receiving Data with FIFO Enabled



63.7.11.4 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_US_CR.TXFCLR/RXFCLR.

63.7.11.5 TXEMPTY, TXRDY and RXRDY Behavior

FLEX_US_CSR.TXEMPTY, FLEX_US_CSR.TXRDY and FLEX_US_CSR.RXRDY flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TXRDY indicates if a data can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new data. See figure TXRDY in Single Data Mode and TXRDYM = 0.

RXRDY indicates if an unread data is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread data is in the Receive FIFO. See figure RXRDY in Single Data Mode and RXRDYM = 0 below.

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the USART FIFO Mode register (FLEX_US_FMR) .

See FLEX_US_FMR for the FIFO configuration.

Figure 63-63. TXRDY Behavior for Single Data Access and TXRDYM = 0

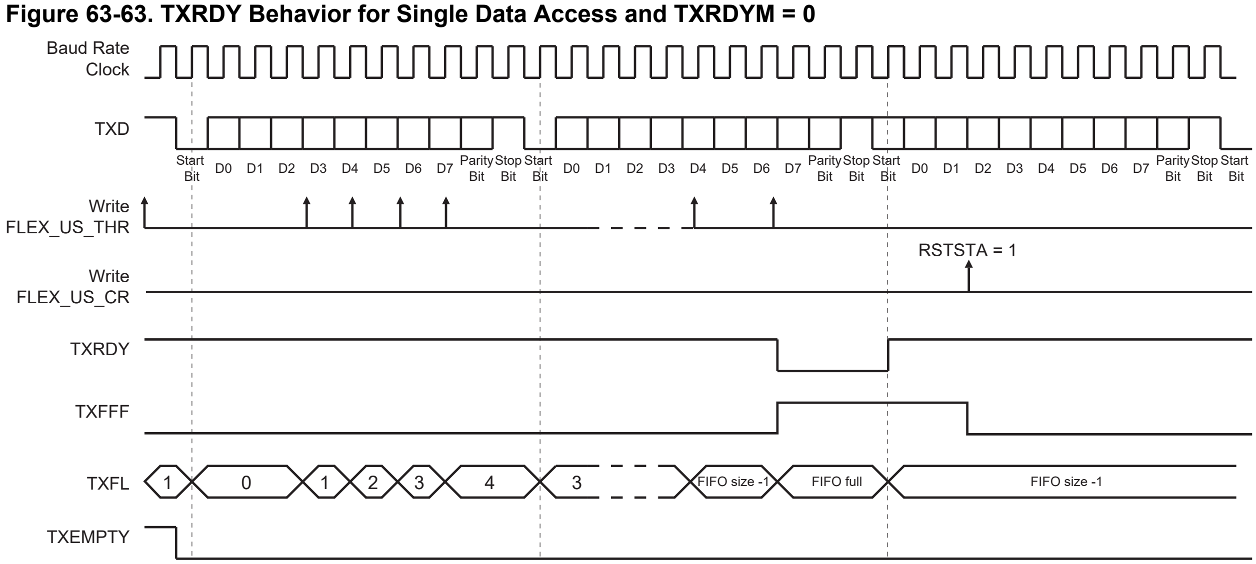
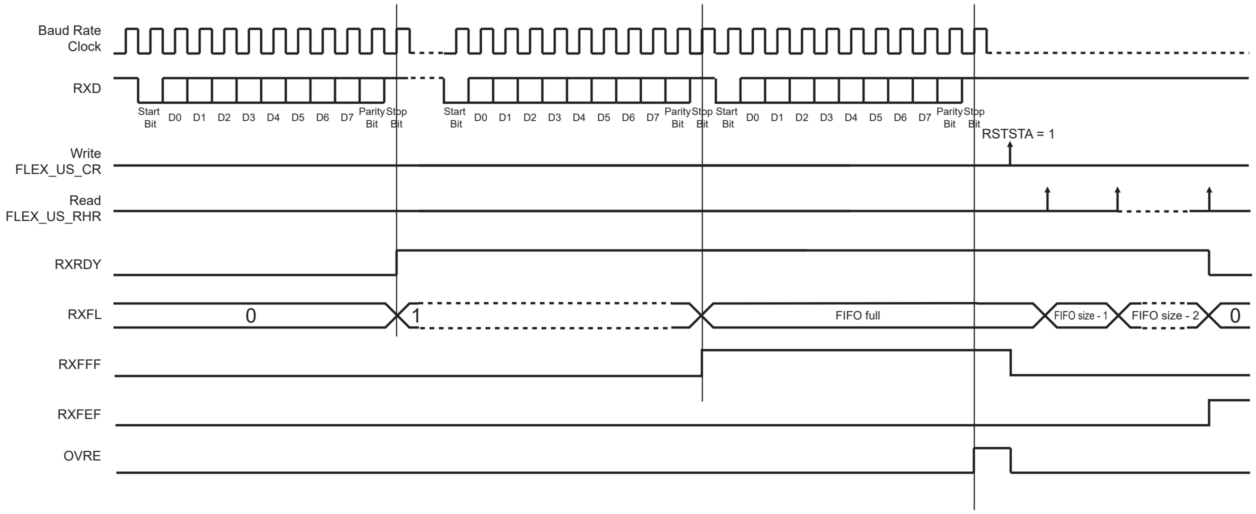


Figure 63-64. RXRDY Behavior for Single Data Access and RXRDYM = 0



63.7.11.6 FIFO Single Data Access

When FIFO is enabled and a byte access is performed in FLEX_US_THR (5-bit to 8-bit data size), a single data is written in FIFO. The similar behavior applies for FLEX_US_RHR.

If FLEX_US_MR.MODE9 is set (9-bit data), or if FLEX_US_MR.USART_MODE is configured to operate in LIN Host mode or LIN Client mode, or if FLEX_US_MR.MAN is set, any type of access to FLEX_US_THR/RHR writes/reads a single data.

See USART Receive Holding Register (FLEX_US_RHR) and USART Transmit Holding Register (FLEX_US_THR).

However, for some configurations it is possible to write/read multiple data each time FLEX_US_THR/FLEX_US_RHR is accessed. See FIFO Multiple Data Access.

63.7.11.6.1 DMAC

The DMAC transfer type must be configured in bytes or halfwords when FIFOs operate in Single Data mode (the same applies when FIFOs are disabled).

63.7.11.7 FIFO Multiple Data Access

For some operating modes, it is possible to reduce the number of accesses to/from FLEX_US_THR/FLEX_US_RHR required to transfer an amount of data, by concatenating multiple data (5-bit to 8-bit) when FIFO is enabled (FLEX_US_CR.FIFOEN=1) and 5- to 8-bit data characters are transferred (FLEX_US_MR.MODE9=0).

Up to four data (5-bit to 8-bit) can be written/read in one FLEX_US_THR/FLEX_US_RHR access.

When the FIFO is enabled, the number of data to write/read is defined by the type of access in the holding register. If the access is a byte, only one data is written/read (single data access), if the access is a halfword or a word a multiple data access is performed. If the access is a halfword, then two data are written/read and if the access is a word, four data are written/read.

Written/read data are always right-aligned, as described in USART Receive Holding Register (FIFO Multi Data) and USART Transmit Holding Register (FIFO Multi Data).

Multiple data access cannot be used for the following configurations:

- If FLEX_US_MR.MODE9 is set
- If FLEX_US_MR.USART_MODE is configured to operated in LIN Host mode or LIN Client mode
- FLEX_US_MR.MAN is set

As an example of multiple data access, if the Transmit FIFO is empty and there are six data to send, any of the following write accesses may be performed:

- six FLEX_US_THR-byte write accesses
- three FLEX_US_THR-halfword write accesses
- one FLEX_US_THR word write access and one FLEX_US_THR halfword write access

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_US_RHR-byte read accesses
- three FLEX_US_RHR-halfword read accesses
- one FLEX_US_RHR-word read access and one FLEX_US_RHR-halfword read access

63.7.11.7.1 TXRDY and RXRDY Configuration

The TXRDY flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_US_FMR.TXRDYM/RXRDYM.

As an example, if a word (32-bit) is written in FLEX_US_THR, the TXRDYM field must be configured so that the TXRDY flag is at ‘1’ only when at least four data can be written in the Transmit FIFO.

In the same way, if a word (32-bit) is read in FLEX_US_RHR, the RXRDYM field must be configured so that the RXRDY flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

63.7.11.7.2 DMAC

The DMAC transfer type must be configured according to the FLEX_US_FMR.TXRDYM/RXRDYM settings.

As an example, FLEX_US_FMR.TXRDYM/RXRDYM=0 is not compatible with DMAC_PDC transfers in word (32-bit).

63.7.11.8 Transmit FIFO Lock

- LIN Mode:

If a frame is aborted using the Abort LIN Transmission bit (FLEX_US_CR.LINABT), a lock is set on the Transmit FIFO, preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMAC channels, etc., without any risk.

The TXFLOCK bit in the USART FIFO Event Status register (FLEX_US_FESR) is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared by setting FLEX_US_CR.TXFLCLR to ‘1’.

63.7.11.9 FIFO Pointer Error

A FIFO overflow is reported in FLEX_US_FESR.

If the Transmit FIFO is full and a write access is performed on FLEX_US_THR, it generates a Transmit FIFO pointer error and sets FLEX_US_FESR.TXFPTEF.

If the number of data written in FLEX_US_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_US_FESR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_US_FESR.

If the number of data read in FLEX_US_RHR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_US_FESR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_US_THR/FLEX_US_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit pointer error occurs, a transmitter reset must be performed using FLEX_US_CR.RSTTX. If a Receive pointer error occurs, a receiver reset must be performed using FLEX_US_CR.RSTRX.

63.7.11.10 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_US_FMR.TXFTHRES. Each time the Transmit FIFO level goes from ‘above threshold’ to ‘equal to or below threshold’, the flag FLEX_US_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_US_FMR.RXFTHRES. Each time the Receive FIFO level goes from ‘below threshold’ to ‘equal to or above threshold’, the flag FLEX_US_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The Receive FIFO threshold 2 can be set using the field FLEX_US_FMR.RXFTHRES2. Each time the Receive FIFO level goes from 'above threshold 2' to 'equal to or below threshold 2', the flag FLEX_US_FESR.RXFTHF2 is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF, RXFTHF and RXTHF2 flags can be configured to generate an interrupt using FLEX_US_FIER and FLEX_US_FIDR.

63.7.11.11 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_US_FIER and FLEX_US_FIDR.

FIFO flags state can be read in FLEX_US_FESR. They are cleared by writing FLEX_US_CR.RSTSTA to '1'.

63.7.12 16-bit Data Protocol Support

When configuring 0xC in FLEX_US_MR.USART_MODE, the transmitter sends a 16-bit data frame and the receiver expects an 8-bit data frame. The number of stop bits is defined in the field NBSTOP. The transmitter and/or receiver must operate in asynchronous mode (FLEX_US_MR.SYNC must be cleared).

When configuring 0xD in FLEX_US_MR.USART_MODE, the transmitter sends an 8-bit frame whereas the receiver expects a 16-bit frame.

The FIFO mode must be enabled by setting FLEX_US_CR.FIFOEN to '1'.

A 16-bit frame starts as soon as two 8-bit characters are written in FLEX_US_THR (assuming 0xC is written in the field USART_MODE).

Note: When FLEX_US_MR.USART_MODE =0xC or 0xD, there must be a parity bit in the frame (FLEX_US_MR.USART_MODE must not be equal to 4).

63.7.13 USART Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_USART to enable access to the write protection registers.

To prevent any single software error from corrupting USART behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the USART Write Protection Mode Register (FLEX_US_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the USART Write Protection Status Register (FLEX_US_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_US_WPSR.

The following registers can be write-protected when WPEN is set:

- USART Mode Register
- USART Baud Rate Generator Register
- USART Receiver Timeout Register
- USART Transmitter Timeguard Register
- USART FI DI RATIO Register
- USART IrDA FILTER Register
- USART Manchester Configuration Register
- USART Comparison Register

The following register(s) can be write-protected when WPITEN is set:

- USART Interrupt Enable Register
- USART Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- USART Control Register

63.8 SPI Functional Description

63.8.1 Modes of Operation

The SPI operates in Host mode or in Client mode.

- The SPI operates in Host mode by writing a 1 to the MSTR bit in the SPI Mode register (FLEX_SPI_MR):
  - The pins NPCS0 to NPCS3 are all configured as outputs.
  - The SPCK pin is driven.
  - The MISO line is wired on the receiver input.
  - The MOSI line is driven as an output by the transmitter.
- The SPI operates in Client mode if the MSTR bit in FLEX_SPI_MR is written to 0:
  - The MISO line is driven by the transmitter output.
  - The MOSI line is wired on the receiver input.
  - The SPCK pin is driven by the transmitter to synchronize the receiver.
  - The NPCS0 pin becomes an input, and is used as a client select signal (NSS).
  - Pins NPCS1 to NPCS3 are not driven and can be used for other purposes.

The data transfers are identically programmable for both modes of operation. The bit rate generator is activated only in Host mode.

63.8.2 Data Transfer

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the SPI Chip Select register (FLEX_SPI_CSR). The clock phase is programmed with the NCPHA bit. These two parameters determine the edges of the clock signal on which data are driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Consequently, a host/client pair must use the same parameter pair values to communicate. If multiple clients are connected and require different configurations, the host must reconfigure itself each time it needs to communicate with a different client.

The following table shows the four modes and corresponding parameter settings.

Table 63-14. SPI Bus Protocol Mode

SPI Mode	CPOL	NCPHA	Host Shift SPCK Edge	Client Capture SPCK Edge	SPCK Inactive Level
0	0	1	Falling	Rising	Low
1	0	0	Rising	Falling	Low
2	1	1	Rising	Falling	High
3	1	0	Falling	Rising	High

The following figures show examples of data transfers.

Figure 63-65. SPI Transfer Format (NCPHA = 1, 8 bits per transfer) Mode 0 and 2

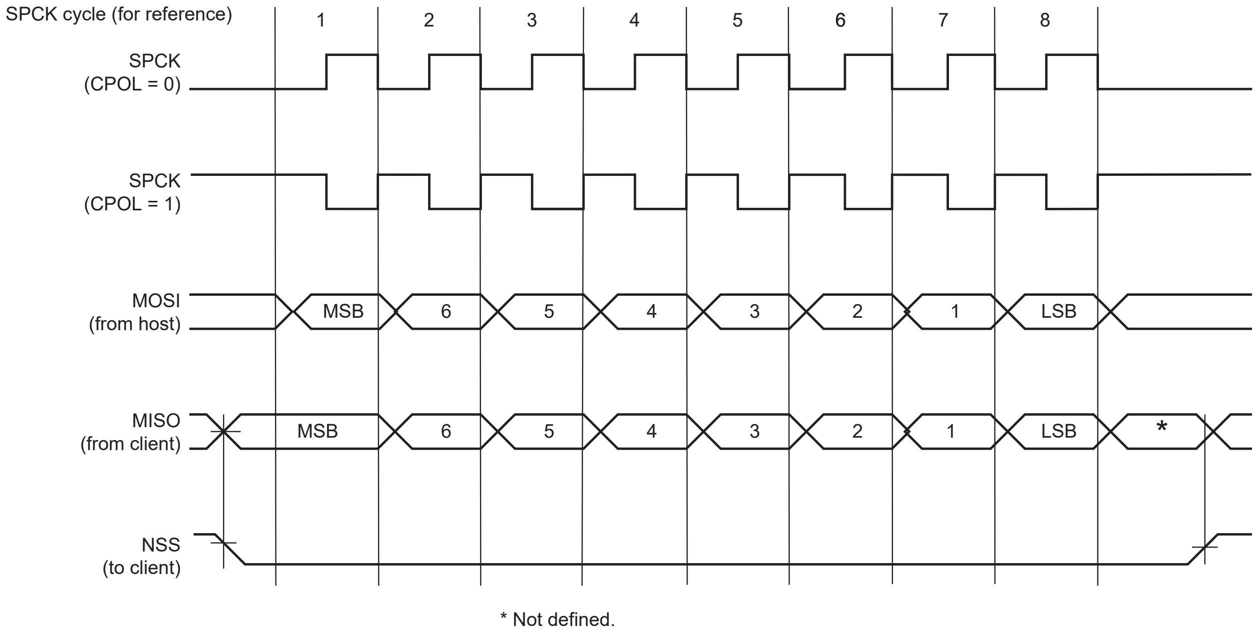
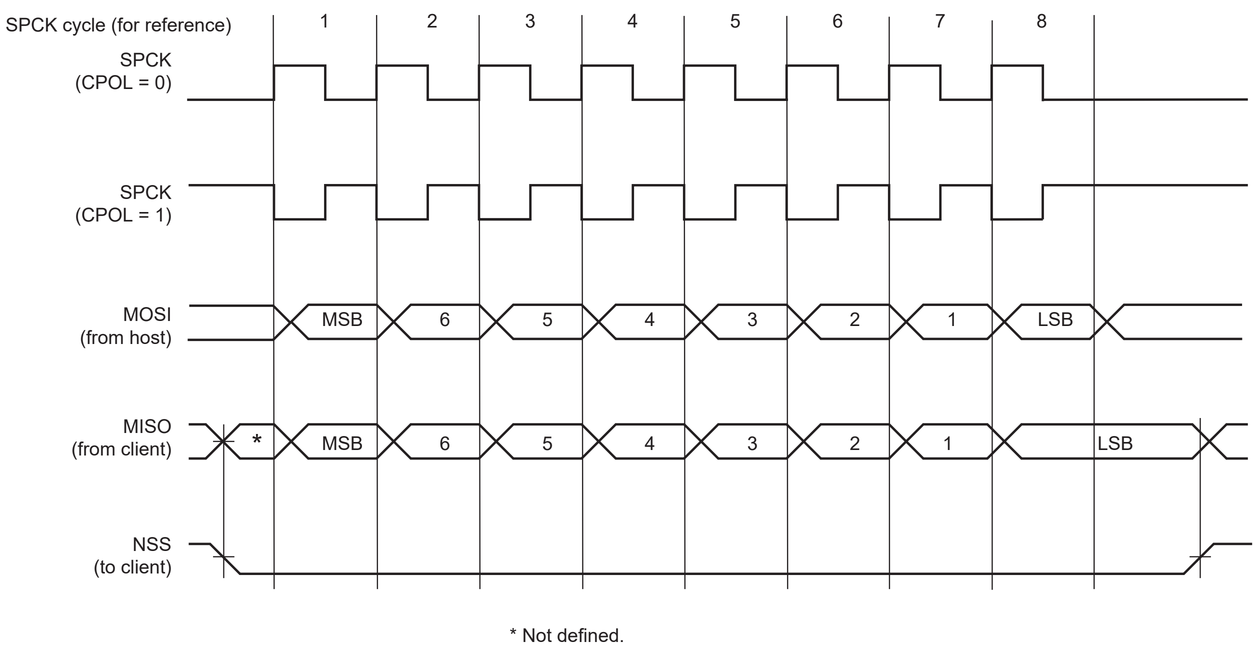


Figure 63-66. SPI Transfer Format (NCPHA = 0, 8 bits per transfer) Mode 1 and 3



63.8.3 Host Mode Operations

When configured in Host mode, the SPI operates on the clock generated by the internal programmable bit rate generator. It fully controls the data transfers to and from the client(s) connected to the SPI bus. The SPI drives the chip select line to the client and the serial clock signal (SPCK).

The SPI features two holding registers, the Transmit Data register (FLEX_SPI_TDR) and the Receive Data register (FLEX_SPI_RDR), and a single shift register. The holding registers maintain the data flow at a constant rate.

After enabling the SPI, a data transfer starts when the processor writes to FLEX_SPI_TDR. The written data are immediately transferred in the shift register and the transfer on the SPI bus starts. While the data in the shift register is shifted on the MOSI line, the MISO line is sampled and shifted in the shift register. Data cannot be loaded in FLEX_SPI_RDR without transmitting data. If there is no data to transmit, a dummy data can be used (FLEX_SPI_TDR filled with ones). When the WDRBT bit is set, a new data cannot be transmitted if FLEX_SPI_RDR has not been read. If Receiving mode is not required, for example when communicating with a client receiver only (such as an LCD), the receive status flags in the SPI Status register (FLEX_SPI_SR) can be discarded.

Before writing the TDR, the FLEX_SPI_MR.PCS field must be set in order to select a client.

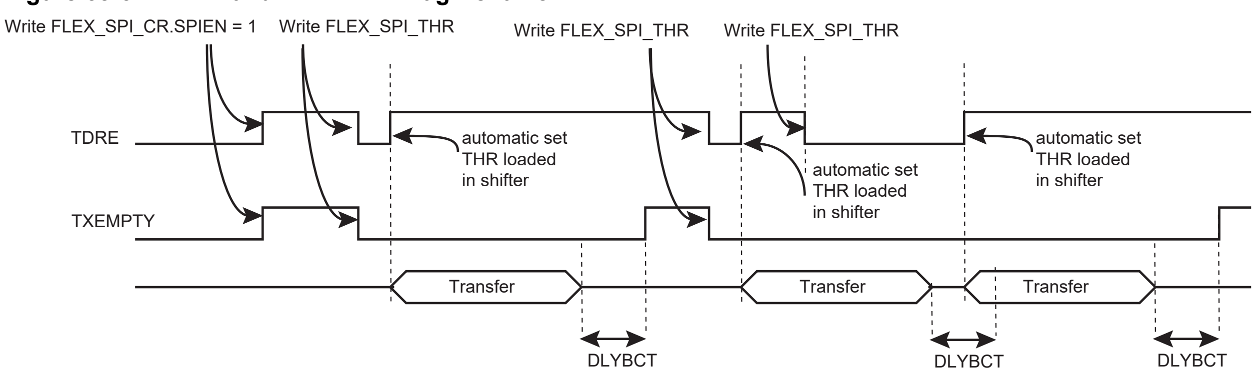
If new data are written in FLEX_SPI_TDR during the transfer, it is kept in FLEX_SPI_TDR until the current transfer is completed. Then, the received data are transferred from the shift register to FLEX_SPI_RDR, the data in FLEX_SPI_TDR is loaded in the shift register and a new transfer starts.

As soon as the FLEX_SPI_TDR is written, the Transmit Data Register Empty (TDRE) flag in FLEX_SPI_SR is cleared. When the data written in FLEX_SPI_TDR is loaded into the shift register, the FLEX_SPI_SR.TDRE flag is set. The TDRE bit is used to trigger the Transmit DMA channel (see figure below).

The end of transfer is indicated by FLEX_SPI_SR.TXEMPTY. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

Notes:
1. When the SPI is enabled, the TDRE and TXEMPTY flags are set.
2. The TXEMPTY flag alone cannot be used to detect the end of the buffer DMA transfer.

Figure 63-67. TDRE and TXEMPTY Flag Behavior



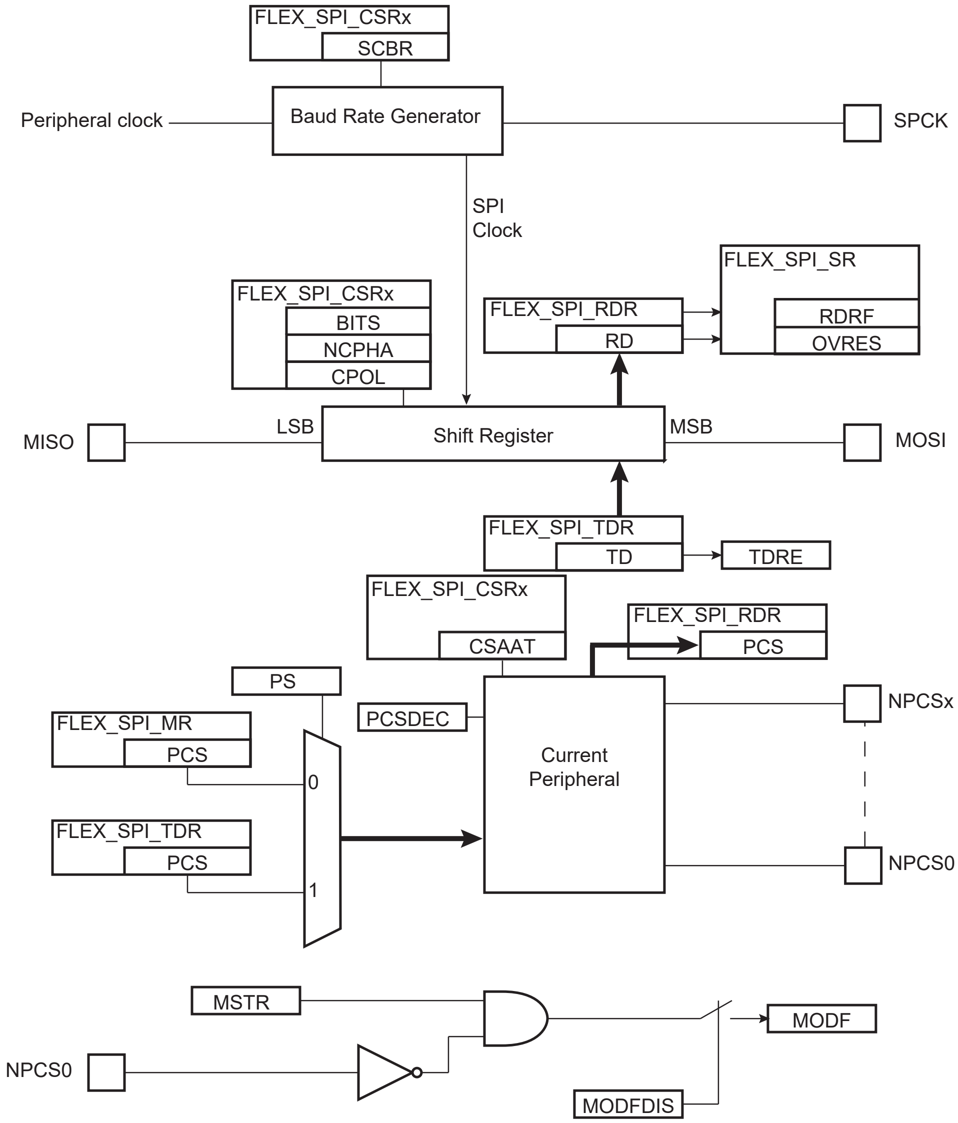
The transfer of received data from the shift register to FLEX_SPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in FLEX_SPI_SR. When the received data are read, the RDRF bit is cleared.

If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user has to read the status register to clear the OVRES bit.

The following figures show, respectively, a block diagram of the SPI when operating in Host mode and a flow chart describing how transfers are handled.

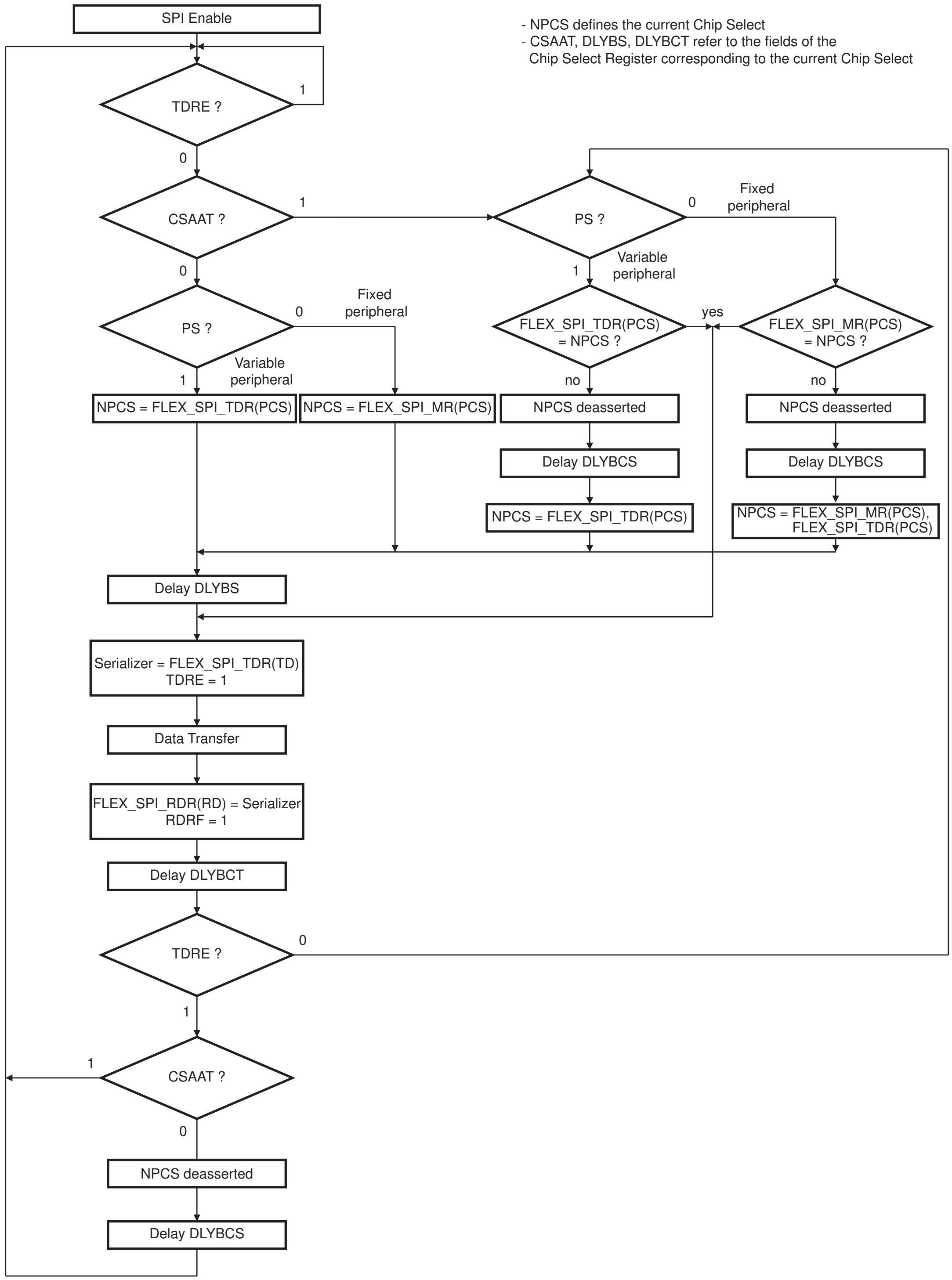
63.8.3.1 Host Mode Block Diagram

Figure 63-68. Host Mode Block Diagram



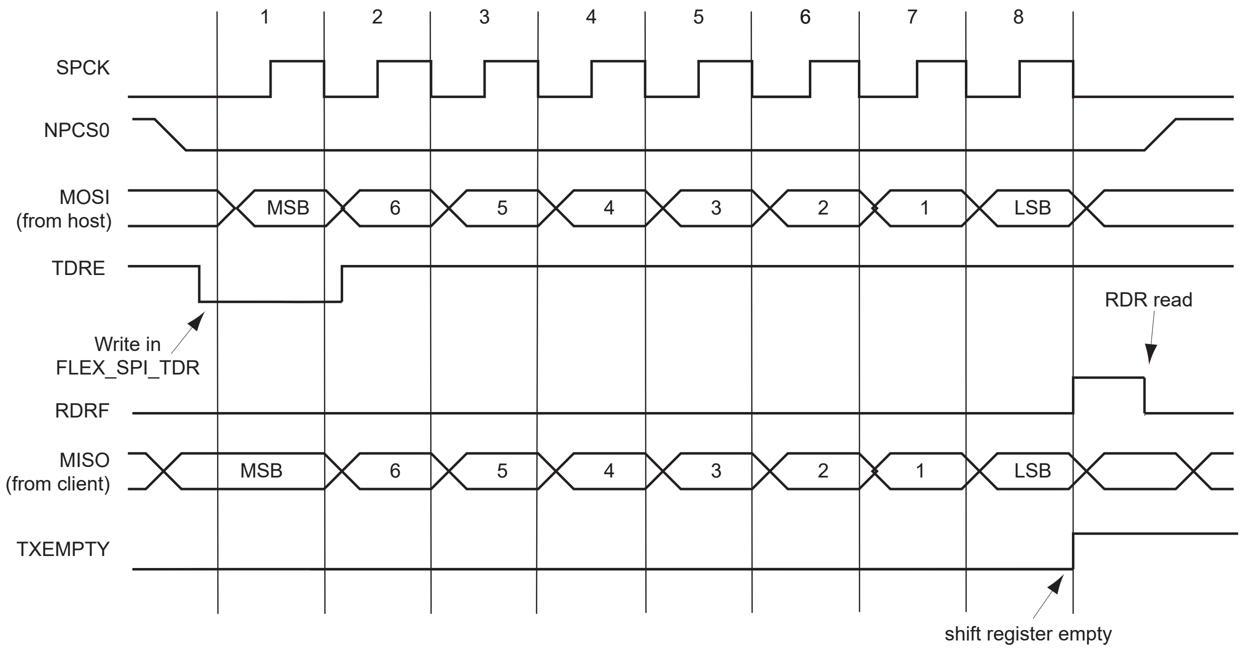
63.8.3.2 Host Mode Flowchart

Figure 63-69. Host Mode



The following figure shows the behavior of Transmit Data Register Empty (TDRE), Receive Data Register (RDRF) and Transmission Register Empty (TXEMPTY) status flags within FLEX_SPI_SR during an 8-bit data transfer in Fixed mode without the DMAC involved.

Figure 63-70. Status Register Flags Behavior



63.8.3.3 Clock Generation

The SPI bit rate clock is generated by dividing a source clock which can be the peripheral clock or a programmable clock from the GCLK. The divider can be a value between 1 and 255.

If the SCBR field is programmed to 1 and the clock source is GCLK, the operating bit rate is peripheral clock (refer to the section “Electrical Characteristics” for the SPCK maximum frequency). Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

At reset, SCBR is 0 and the user has to program it to a valid value before performing the first transfer.

The divisor can be defined independently for each chip select, as it has to be programmed in the FLEX_SPI_CSR.SCBR field. This allows the SPI to automatically adapt the bit rate for each interfaced peripheral without reprogramming.

If GCLK is selected as source clock (FLEX_SPI_MR.BRSRCCLK = 1), the bit rate is independent of the processor/bus clock. Thus, the processor clock can be changed while SPI is enabled. The processor clock frequency changes must be performed only by programming the PMC_MCKR.PRES field (refer to the section “Power Management Controller (PMC)"). Any other method to modify the processor/bus clock frequency (PLL multiplier, etc.) is forbidden when SPI is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

63.8.3.4 Transfer Delays

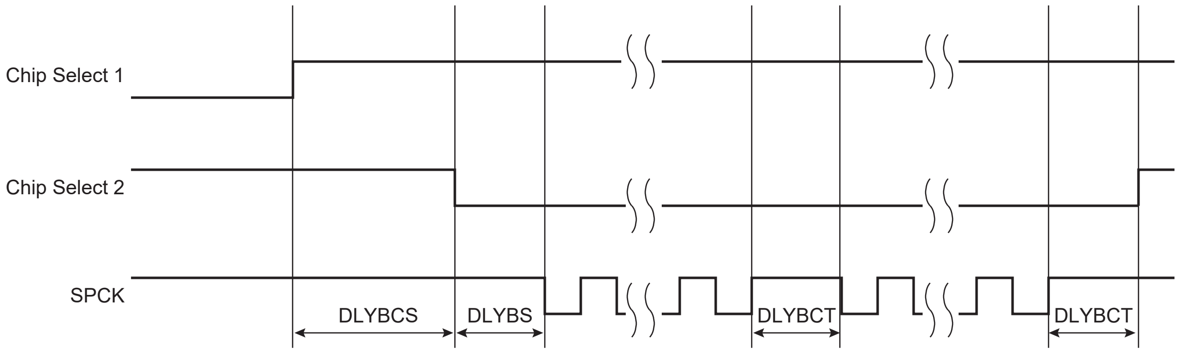
The figure below shows a chip select transfer change and consecutive transfers on the same chip select. Three delays can be programmed to modify the transfer waveforms:

- The delay between the chip selects. It is programmable only once for all chip selects by writing the FLEX_SPI_MR.DLYBCS field. The SPI client device deactivation delay is managed through DLYBCS. If there is only one SPI client device connected to the host, the DLYBCS field does not need to be configured. If several client devices are connected to a host, DLYBCS must be configured depending on the highest deactivation delay. Refer to the SPI client device electrical characteristics.
- The delay before SPCK, independently programmable for each chip select by writing the DLYBS field. The SPI client device activation delay is managed through DLYBS. Refer to the SPI client device electrical characteristics to define DLYBS.

- The delay between consecutive transfers, independently programmable for each chip select by writing the DLYBCT field. The time required by the SPI client device to process received data is managed through DLYBCT. This time depends on the SPI client system activity.

These delays allow the SPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 63-71. Programmable Delays



63.8.3.5 Peripheral Selection

The serial peripherals are selected through the assertion of the NPCS0 to NPCS3 signals. By default, all NPCS signals are high before and after each transfer.

- Fixed Peripheral Select Mode: SPI exchanges data with only one peripheral.
  Fixed Peripheral Select mode is enabled by writing the FLEX_SPI_MR.PS bit to zero. In this case, the current peripheral is defined by the FLEX_SPI_MR.PCS field, and the FLEX_SPI_TDR. PCS field has no effect.
- Variable Peripheral Select Mode: Data can be exchanged with more than one peripheral without having to reprogram FLEX_SPI_MR.PCS.
  Variable Peripheral Select Mode is enabled by setting the FLEX_SPI_MR.PS bit to one. The FLEX_SPI_TDR.PCS field is used to select the current peripheral. This means that the peripheral selection can be defined for each new data. The value must be written in a single access to FLEX_SPI_TDR in the following format:

  [xxxxxxx(7-bit) + LASTXFER(1-bit)[1]+ xxxx(4-bit) + PCS (4-bit) + TD (8 to 16-bit data)]
  with LASTXFER at 0 or 1 depending on the CSAAT bit, and PCS equal to the chip select to assert, as defined in SPI Transmit Data Register (FLEX_SPI_TDR).
  Note: 1. Optional

  The CSAAT, LASTXFER and CSNAAT bits are discussed in Peripheral Deselection with DMA.

  If LASTXFER is used, the command must be issued after writing the last character. Instead of LASTXFER, the user can use the SPIDIS command. After the end of the DMA transfer, it is necessary to wait for the TXEMPTY flag and then write SPIDIS into the SPI Control register (FLEX_SPI_CR). This does not change the configuration register values). The NPCS is disabled after the last character transfer. Then, another DMA transfer can be started if the FLEX_SPI_CR.SPIEN bit has previously been written.

63.8.3.6 SPI Direct Access Memory Controller (DMAC)

In both Fixed and Variable modes, the Direct Memory Access Controller (DMAC) can be used to reduce processor overhead.

The fixed peripheral selection allows buffer transfers with a single peripheral. Using the DMAC is an optimal means, as the size of the data transfer between the memory and the SPI is either 8 bits or 16 bits. However, if the peripheral selection is modified, FLEX_SPI_MR must be reprogrammed.

The variable peripheral selection allows buffer transfers with multiple peripherals without reprogramming FLEX_SPI_MR. Data written in FLEX_SPI_TDR is 32 bits wide and defines the real data to be transmitted and the destination peripheral. Using the DMAC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs. However, the SPI still controls the number of bits (8 to 16) to be transferred through MISO and MOSI lines with the chip select configuration registers. This is not the optimal means in terms of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

63.8.3.7 Peripheral Chip Select Decoding

The user can program the SPI to operate with up to 15 client peripherals by decoding the four chip select lines, NPCS0 to NPCS3 with an external decoder/demultiplexer (see the following figure). This can be enabled by setting the FLEX_SPI_MR.PCSDEC bit.

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

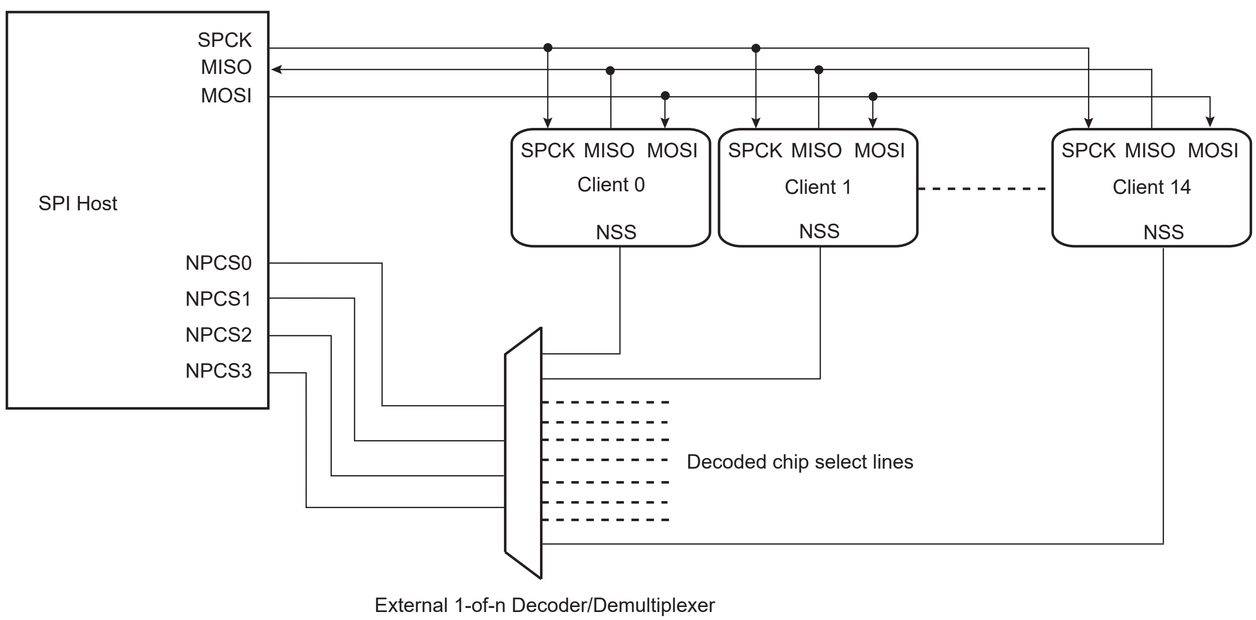
When operating with decoding, the SPI directly outputs the value defined by the PCS field on the NPCS lines of either FLEX_SPI_MR or FLEX_SPI_TDR (depending on PS).

As the SPI sets a default value of 0xF on the chip select lines (i.e., all chip select lines at 1) when not processing any transfer, only 15 peripherals can be decoded.

The SPI has only four Chip Select registers. As a result, when external decoding is activated, each NPCS chip select defines the characteristics of up to four peripherals. As an example, FLEX_SPI_CRS0 defines the characteristics of the externally decoded peripherals 0 to 3, corresponding to the PCS values 0x0 to 0x3. Consequently, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 3, 4 to 7, 8 to 11 and 12 to 14. The following figure shows this type of implementation.

If the CSAAT bit is used, with or without the DMAC, the mode fault detection for NPCS0 line must be disabled. This is not needed for all other chip select lines since mode fault detection is only on NPCS0.

Figure 63-72. Chip Select Decoding Application Block Diagram: Single Host/Multiple Client Implementation



63.8.3.8 Peripheral Deselection without DMA

During a transfer of more than one data on a Chip Select without the DMA, FLEX_SPI_TDR is loaded by the processor, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected high, FLEX_SPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. But depending on the application software handling the SPI status register flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload FLEX_SPI_TDR in time to keep the chip select active (low). A null DLYBCT value (delay between consecutive transfers) in FLEX_SPI_CSR, gives even less time for the processor to reload FLEX_SPI_TDR. With some SPI client peripherals, if the chip select line must remain active (low) during a full set of transfers, communication errors can occur.

To facilitate interfacing with such devices, the Chip Select registers [CSR0...CSR3] can be programmed with the Chip Select Active After Transfer (CSAAT) bit to 1. This allows the chip select lines to remain in their current state (low = active) until a transfer to another chip select is required. Even if FLEX_SPI_TDR is not reloaded, the chip select remains active. To de-assert the chip select line at the end of the transfer, the Last Transfer (LASTXFER) bit in FLEX_SPI_CR must be set after writing the last data to transmit into FLEX_SPI_TDR.

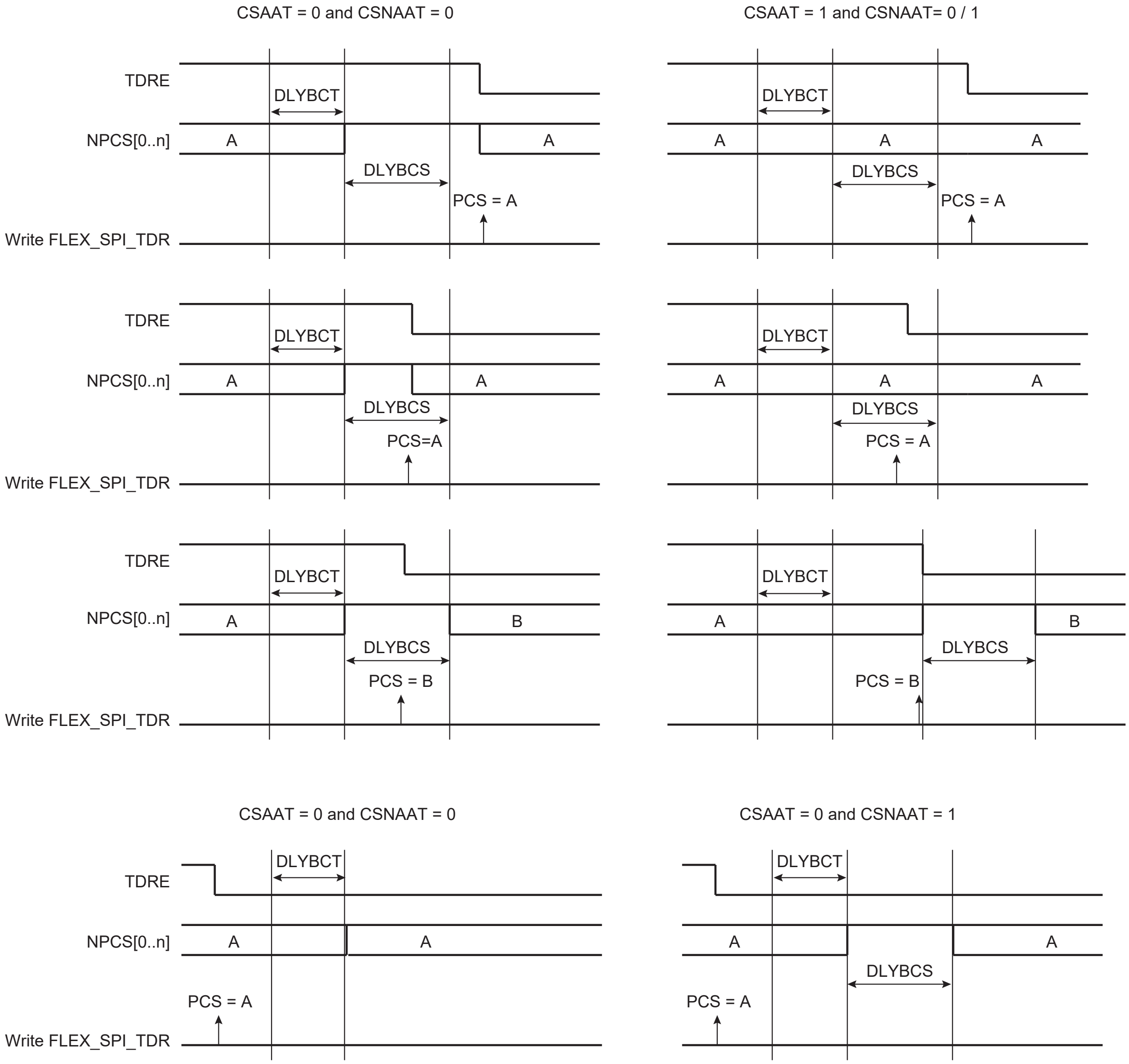
63.8.3.9 Peripheral Deselection with DMA

DMA provides faster reloads of FLEX_SPI_TDR compared to software. However, depending on the system activity, it is never sure that FLEX_SPI_TDR is written with the next data before the end of the current transfer. Consequently, a data can be lost by the deassertion of the NPCS line for SPI client peripherals requiring the chip select line to remain active between two transfers. The only way to ensure a safe transfer in this case is the use of the CSAAT and LASTXFER bits.

When the CSAAT bit is cleared, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected, FLEX_SPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. This can lead to difficulties to interface with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices, FLEX_SPI_CSR can be programmed with the Chip Select Not Active After Transfer (CSNAAT) bit to 1. This allows the chip select lines to be deasserted systematically during a time “DLYBCS” (the value of the CSNAAT bit is processed only if the CSAAT bit is cleared for the same chip select).

The following figure shows different peripheral deselection cases and the effect of the CSAAT and CSNAAT bits.

Figure 63-73. Peripheral Deselection



63.8.3.10 Mode Fault Detection

The SPI has the capability to operate in multi-host environment. Consequently, the NPCS0/NSS line must be monitored. If one of the hosts on the SPI bus is currently transmitting, the NPCS0/NSS line is low and the SPI must not transmit a data. A mode fault is detected when the SPI is programmed in Host mode and a low level is driven by an external host on the NPCS0/NSS signal. In multi-host environment, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the FLEX_SPI_SR.MODF bit is set until FLEX_SPI_SR is read and the SPI is automatically disabled until it is re-enabled by writing the FLEX_SPI_CR.SPIEN bit to 1.

By default, the mode fault detection is enabled. The user can disable it by setting the FLEX_SPI_MR.MODFDIS bit.

63.8.4 SPI Client Mode

When operating in Client mode, the SPI processes data bits on the clock provided on the SPI clock pin (SPCK).

The SPI waits until NSS goes active before receiving the serial clock from an external host. When NSS falls, the clock is validated and the data are loaded in FLEX_SPI_RDR according to the configuration value of the FLEX_SPI_CSR0.BITS field. These bits are processed following a phase and a polarity defined respectively by the

FLEX_SPI_CSR0.NCPHA and FLEX_SPI_CSR0.CPOL bits. Note that the BITS field, CPOL bit and NCPHA bit of the other Chip Select registers have no effect when the SPI is programmed in Client mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

Note: For more information on the BITS field, see also the note below the FLEX_SPI_CSRx register bitmap in section SPI Chip Select Register

When all bits are processed, the received data are transferred in FLEX_SPI_RDR and the RDRF bit rises. If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user must read FLEX_SPI_SR to clear the OVRES bit.

When a transfer starts, the data shifted out is the data present in the shift register. If no data has been written in FLEX_SPI_TDR, the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the shift register resets to 0.

When a first data is written in FLEX_SPI_TDR, it is transferred immediately in the shift register and the TDRE flag rises. If new data is written, it remains in FLEX_SPI_TDR until a transfer occurs, i.e., NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in FLEX_SPI_TDR is transferred in the shift register and the TDRE flag rises. This enables frequent updates of critical variables with single transfers.

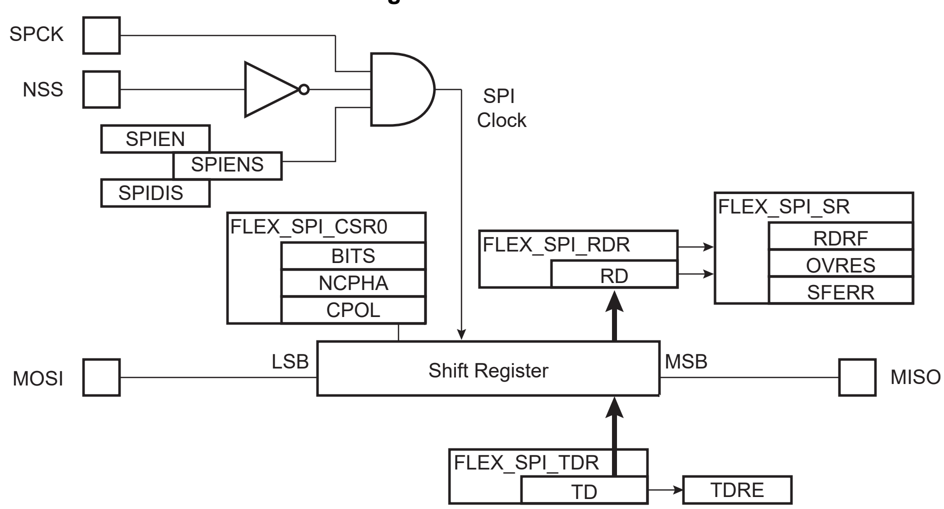
Then, a new data is loaded in the shift register from FLEX_SPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in FLEX_SPI_TDR since the last load from FLEX_SPI_TDR to the shift register, FLEX_SPI_TDR is retransmitted. In this case the Underrun Error Status flag (UNDES) is set in FLEX_SPI_SR.

If NSS rises between two characters, it must be kept high for two MCK clock periods or more and the next SPCK capture edge must not occur less than four MCK periods after NSS rise.

In Client mode, if the NSS line rises and the received character length does not match the configuration defined in FLEX_SPI_CSR0.BITS, the SFERR flag is set in FLEX_SPI_SR.

The following figure shows a block diagram of the SPI when operating in Client mode.

Figure 63-74. Client Mode Functional Block Diagram



63.8.5 SPI Comparison Function on Received Character

The comparison is only relevant for SPI Client mode (MSTR = 0 in FLEX_US_MR).

The effect of a comparison match changes if the system is in Wait or Active mode.

In Wait mode, if asynchronous partial wakeup is enabled, a system wakeup is performed (see 63.8.6. SPI Asynchronous and Partial Wake-up).

In Active mode, the CMP flag in FLEX_SPI_SR is raised. It is set when the received character matches the conditions programmed in the SPI Comparison register (FLEX_SPI_CMPR). The CMP flag is set as soon as FLEX_SPI_RDR is loaded with the new received character. The CMP flag is cleared by reading FLEX_SPI_SR.

The SPI Comparison register can be programmed to provide different comparison methods. These are listed below:

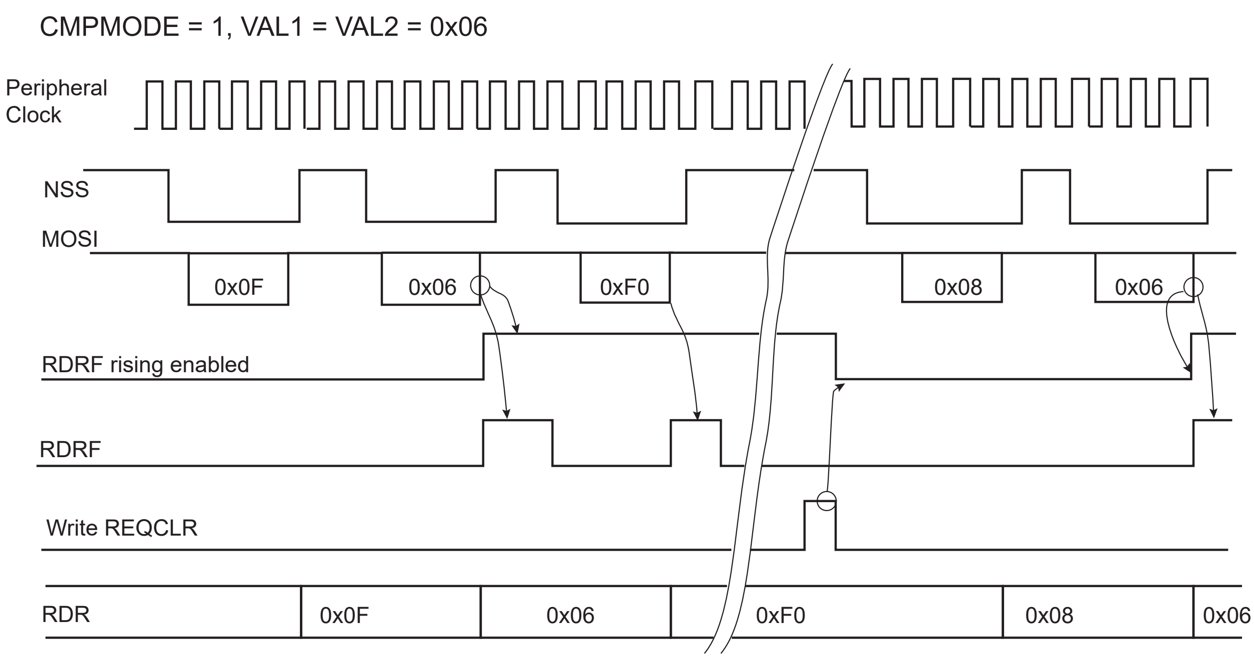
- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When FLEX_SPI_MR.CMPMODE is cleared, all received data is loaded in FLEX_SPI_RDR and the CMP flag provides the status of the comparison result.

By setting the CMPMODE bit, the comparison result triggers the start of FLEX_SPI_RDR loading (see the figure below). The trigger condition exists as soon as the received character value matches the conditions defined by VAL1 and VAL2 in FLEX_SPI_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_SPI_CR.REQCLR bit.

The value programmed in VAL1 and VAL2 fields must not exceed the maximum value of the received character (see BITS field in SPI Chip Select register (FLEX_SPI_CSR)).

Figure 63-75. Receive Data Register Management



63.8.6 SPI Asynchronous and Partial Wake-up

This operating mode is a means of data preprocessing that qualifies an incoming event, thus allowing the SPI to decide whether or not to wake up the system. Asynchronous and partial wakeup is mainly used when the system is in Wait mode (refer to the section “Power Management Controller (PMC)" for further details). It can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

Asynchronous and partial wake-up can be used only when SPI is configured in Client mode (FLEX_SPI_MR.MSTR is cleared).

The maximum SPI clock (SPCK) frequency that can be provided by the SPI host is bounded by the peripheral clock frequency. The SPCK frequency must be lower than or equal to the peripheral clock. The NSS line must be deasserted by the SPI host between two characters. The NSS deassertion duration time must be greater than or equal to six peripheral clock periods. The time between the assertion of NSS line (falling edge) and the first edge of the SPI clock must be higher than 15 µs.

The FLEX_SPI_RDR register must be read before enabling the asynchronous and partial wake-up.

When asynchronous and partial wake-up is enabled for the SPI (refer to the section “Power Management Controller (PMC)"), the PMC decodes a clock request from the SPI. The request is generated as soon as there is a falling edge on the NSS line as this may indicate the beginning of a frame. If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the SPI.

The SPI processes the received frame and compares the received character with VAL1 and VAL2 in FLEX_SPI_CMPR.

The SPI instructs the PMC to disable the peripheral clock if the received character value does not meet the conditions defined by VAL1 and VAL2 fields in FLEX_SPI_CMPR (see figure Asynchronous Wake-up Use Case Example).

If the received character value meets the conditions, the SPI instructs the PMC to exit the system from Wait mode (see figure Asynchronous Event Generating Only Partial Wake-up).

The VAL1 and VAL2 fields can be programmed to provide different comparison methods and thus matching conditions.

- If VAL1 equals VAL2, then the comparison is performed on a single value and the wake-up is triggered if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 wakes up the system.
- If VAL1 is strictly higher than VAL2, the wake-up is triggered if any received character equals VAL1 or VAL2.
- If VAL1 = 0 and VAL2 = 65535, the wake-up is triggered as soon as a character is received.

If the processor and peripherals are running, the SPI can be configured in Asynchronous and Partial Wake-up mode by enabling the PMC_SLPWK_ER (refer to the section “Power Management Controller (PMC)"). When activity is detected on the receive line, the SPI requests the clock from the PMC and the comparison is performed. If there is a comparison match, the SPI continues to request the clock. If there is no match, the clock is switched off for the SPI only, until a new activity is detected.

The CMPMODE configuration has no effect when Asynchronous and Partial Wake-up mode is enabled for the SPI (refer to PMC_SLPWK_ER in the section “Power Management Controller (PMC)").

When the system is in Active mode and the SPI enters Asynchronous and Partial Wake-up mode, the flag RDRF must be programmed as the unique source of the SPI interrupt.

When the system exits Wait mode as the result of a matching condition, the RDRF flag is used to determine if the SPI is the source for the exit from Wait mode.

Figure 63-76. Asynchronous Wake-up Use Case Example

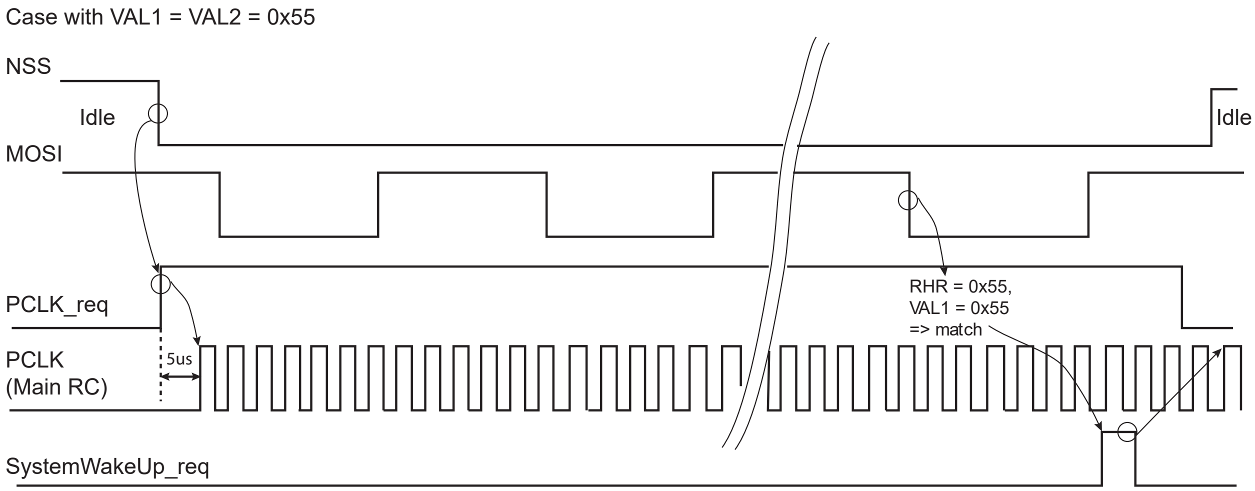
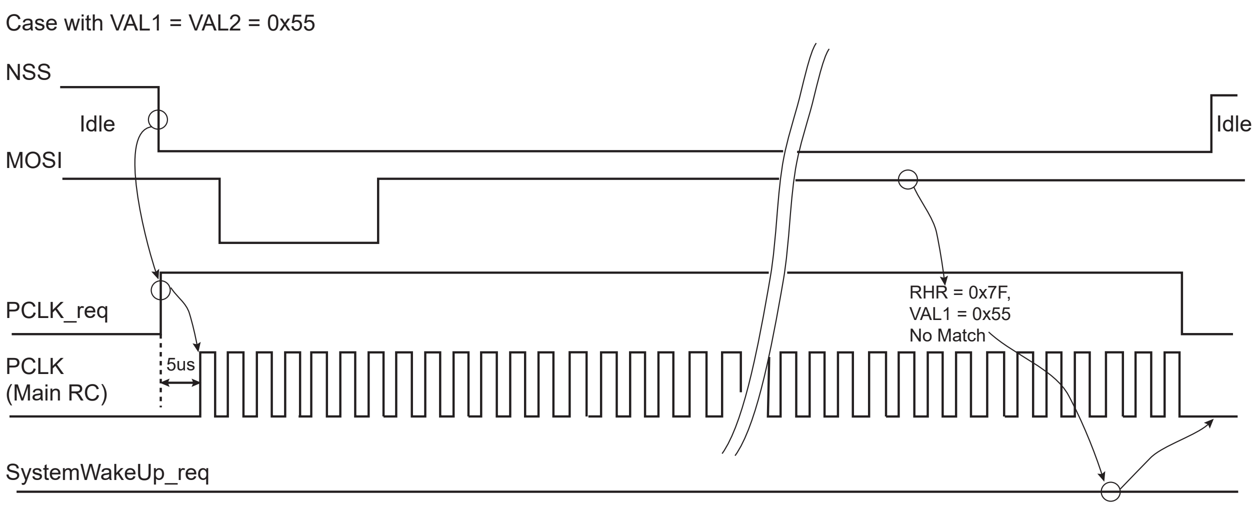


Figure 63-77. Asynchronous Event Generating Only Partial Wake-up



63.8.7 SPI FIFOs

63.8.7.1 Overview

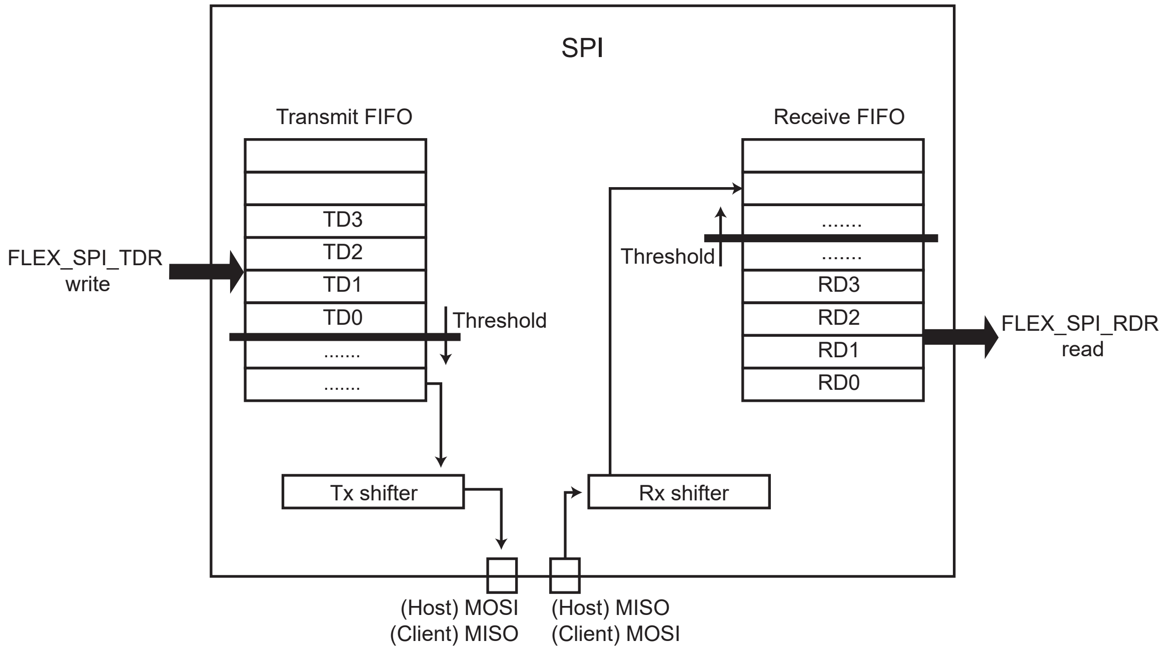
The SPI includes two FIFOs which can be enabled/disabled using the FLEX_SPI_CR.FIFOEN/FIFODIS. The SPI module must be disabled before enabling or disabling the SPI FIFOs (FLEX_SPI_CR.SPIDIS).

Writing FLEX_SPI_CR.FIFOEN to ‘1’ enables a 32-data Transmit FIFO and a 32-data Receive FIFO.

The size of a data (8-bit to 16-bit) is determined by the value configured in FLEX_SPI_CSRx.BITS.

It is possible to write or to read single or multiple data in the same access to FLEX_SPI_TDR/RDR. See SPI Single Data Access and SPI Multiple Data Access.

Figure 63-78. SPI FIFOs Block Diagram



63.8.7.2 Sending Data with FIFO Enabled

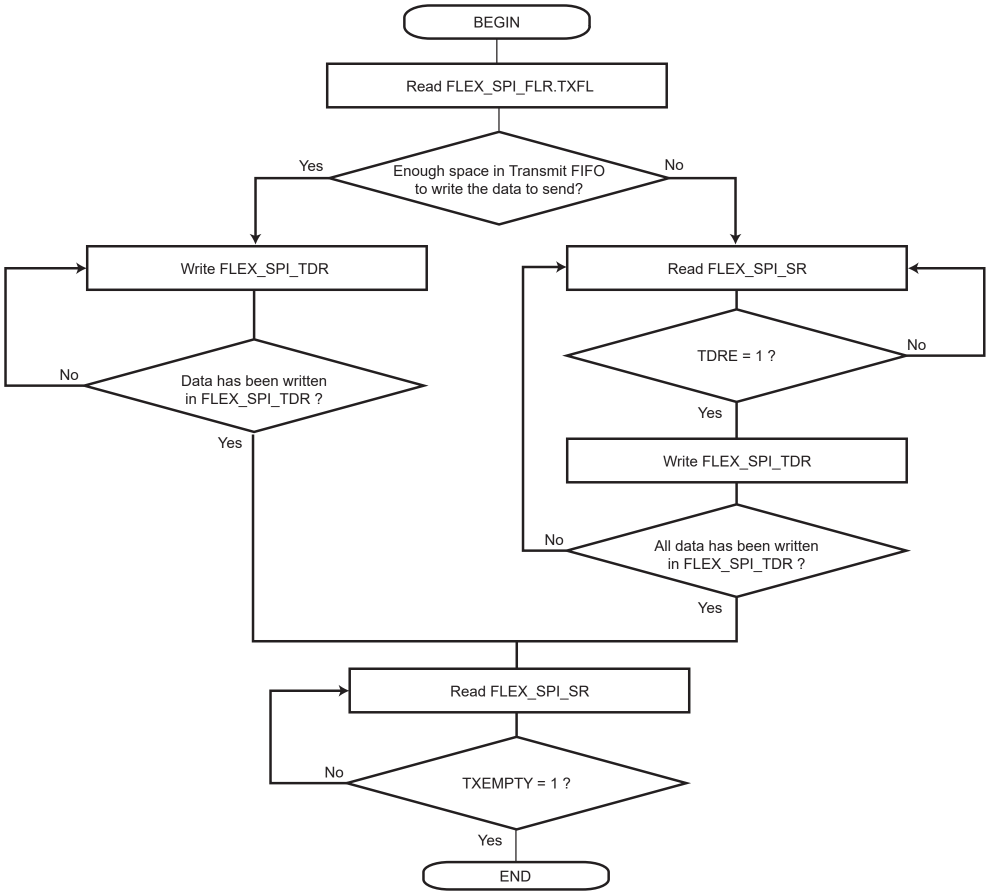
When the Transmit FIFO is enabled, write access to FLEX_SPI_TDR loads the Transmit FIFO.

The FIFO level is provided in FLEX_SPI_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_SPI_SR.TDRE and the data can be successively written in FLEX_SPI_TDR.

If the FIFO cannot accept the data due to insufficient space, wait for the TDRE flag to be set before writing the data in FLEX_SPI_TDR.

When the space in the FIFO allows only a portion of the data to be written, the TDRE flag must be monitored before writing the remaining data.

Figure 63-79. Sending Data with FIFO Enabled

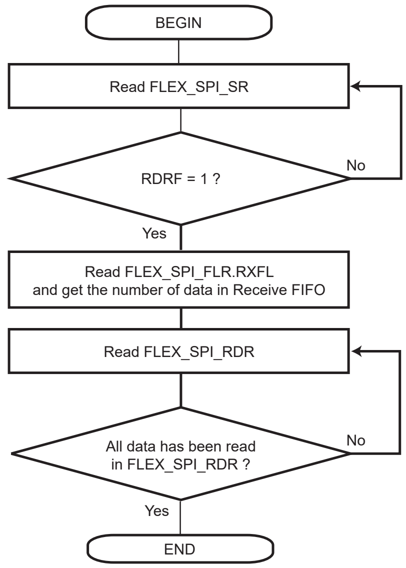


63.8.7.3 Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_SPI_RDR access reads the FIFO.

When data are present in the Receive FIFO (RDRF flag set to ‘1’), the exact number of data can be checked with FLEX_SPI_FLR.RXFL. All the data can be read successively in FLEX_SPI_RDR without checking the RDRF flag between each access.

Figure 63-80. Receiving Data with FIFO Enabled



63.8.7.4 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_SPI_CR.TXFCLR/RXFCLR.

63.8.7.5 TXEMPTY, TDRE and RDRF Behavior

FLEX_SPI_SR.TXEMPTY, FLEX_SPI_SR.TDRE and FLEX_SPI_SR.RDRF flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TDRE indicates if a data can be written in the Transmit FIFO. Thus the TDRE flag is set as long as the Transmit FIFO can accept new data. See figure TDRE Behavior for Single Data Access and TXRDYM = 0.

RDRF indicates if an unread data is present in the Receive FIFO. Thus the RDRF flag is set as soon as one unread data is in the Receive FIFO. See figure RDRF Behavior in Single Data Access and RXRDYM = 0.

TDRE and RDRF behavior can be modified using the TXRDYM and RXRDYM fields in the SPI FIFO Mode register (FLEX_SPI_FMR) to reduce the number of accesses to FLEX_SPI_TDR/RDR. However, for some configurations, the following constraints apply:

- When Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), TXRDYM/RXRDYM must be cleared.
- In Host mode (FLEX_SPI_MR.MSTR=1), RXRDYM must be cleared.

As an example, in Host mode, the Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0.

See SPI FIFO Mode register (FLEX_SPI_FMR) for the FIFO configuration.

Figure 63-81. TDRE Behavior for Single Data Access and TXRDYM = 0

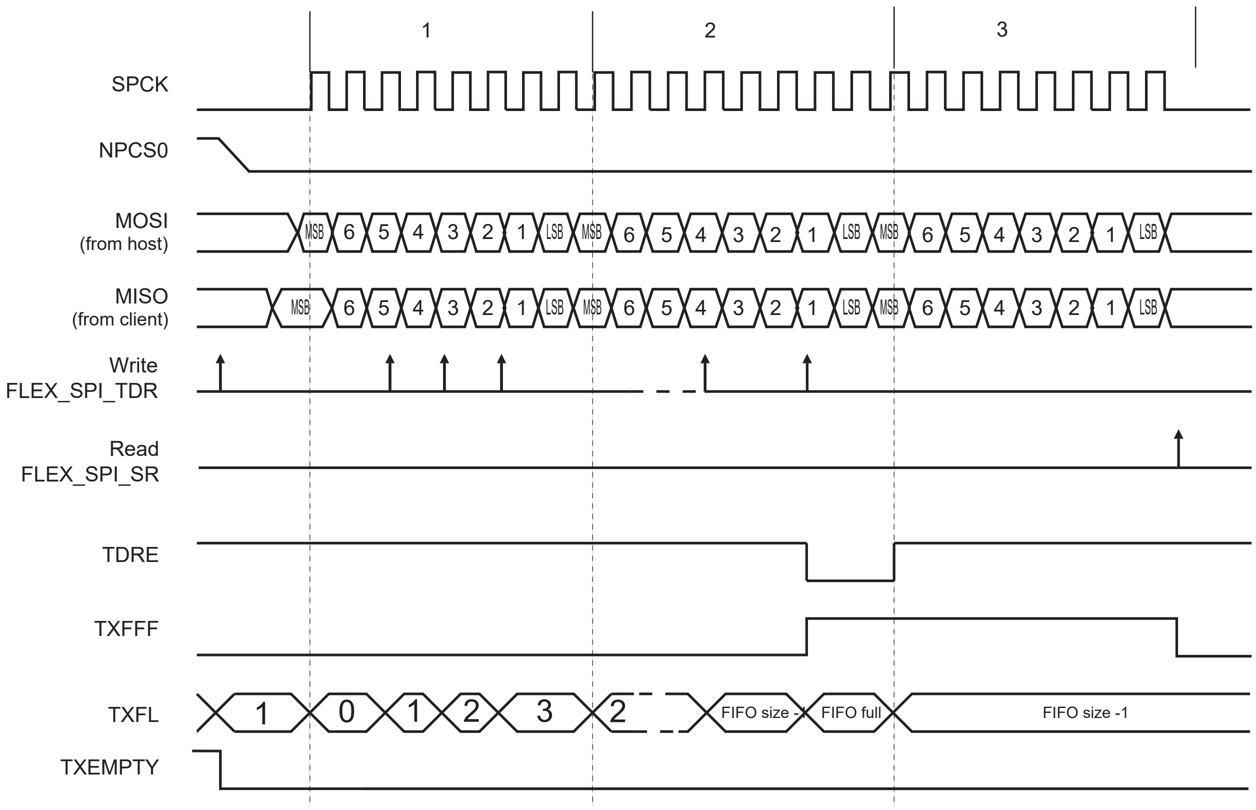
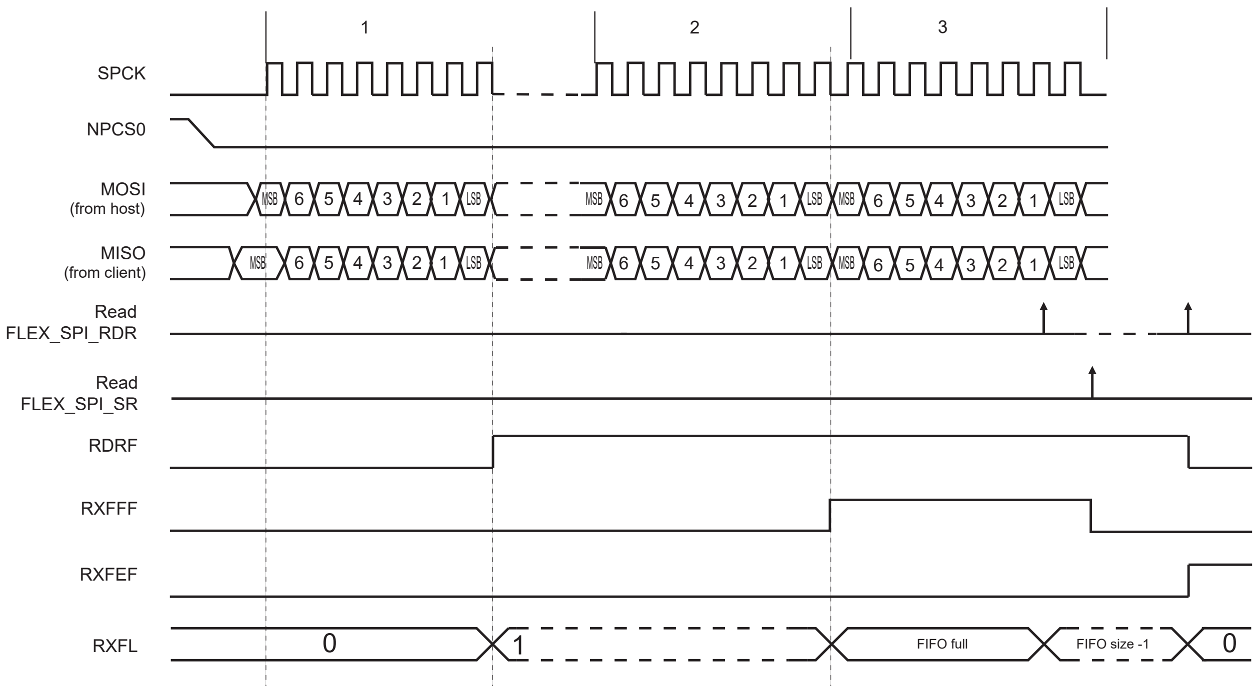


Figure 63-82. RDRF Behavior for Single Data Access and RXRDYM = 0



63.8.7.6 SPI Single Data Access

When FIFO is enabled and a byte or a halfword access (8-bit to 16-bit data) is performed in FLEX_SPI_TDR, a single data is written in FIFO each time FLEX_SPI_TDR is accessed. The similar behavior applies for FLEX_SPI_RDR.

If Host mode is used (FLEX_SPI_MR.MSTR=1) or if Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), each access to FLEX_SPI_RDR must be read a single data.

See SPI Transmit Data Register and SPI Receive Data Register.

However, for some configurations it is possible to write/read multiple data each time FLEX_SPI_TDR/ FLEX_SPI_RDR is accessed. See SPI Multiple Data Access.

63.8.7.6.1 DMAC

When FIFOs operate in Single Data mode, the DMAC transfer type must be configured either in bytes, halfwords or words depending on FLEX_SPI_MR.PS bit value and FLEX_SPI_CSRx.BITS field value.

The same applies when FIFOs are disabled.

63.8.7.7 SPI Multiple Data Access

For some operating modes, it is possible to reduce the number of accesses to FLEX_SPI_TDR/FLEX_SPI_RDR required to transfer an amount of data, by concatenating multiple data (8-bit or 9-bit to 16-bit) when the FIFO is enabled (FLEX_SPI_CR.FIFOEN=1) and fixed peripheral select is used (FLEX_SPI_MR.PS=0).

Up to two data can be written in one FLEX_SPI_TDR write access.

Up to four data can be read in one FLEX_SPI_RDR access.

When the FIFO is enabled, the number of data written in a single access to FLEX_SPI_TDR is only defined by the type of access.

Table 63-15. Number of Data Written for Each Access to FLEX_SPI_TDR

Config/Access	Byte	Halfword	Word
8-bit to 16-bit	1	1	2

When the FIFO is enabled, the number of data read in a single access to FLEX_SPI_RDR is defined by the type of access and the configuration of FLEX_SPI_CSR0.BITS.

Table 63-16. Number of Data Read for Each Access to FLEX_SPI_RDR

Config/Access	Byte	Halfword	Word
FLEX_SPI_CSR0.BITS=0 8-bit data	1	2	4
FLEX_SPI_CSR0.BITS>0 9-bit to 16-bit data	1	1	2

Multiple data can be read from the Receive FIFO only in Client mode (FLEX_SPI_MR.MSTR=0).

The Transmit FIFO can be loaded with multiple data in the same FLEX_SPI_TDR access when FLEX_SPI_MR.PS=0.

Written/read data are always right-aligned, as described in sections SPI Receive Data Register (FIFO Multiple Data, 8-bit), SPI Receive Data Register (FIFO Multiple Data, 16-bit) and SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_SPI_TDR-byte write accesses
- three FLEX_SPI_TDR-halfword write accesses

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_SPI_RDR-byte read accesses
- three FLEX_SPI_RDR-halfword read accesses
- one FLEX_SPI_RDR-word read access and one FLEX_SPI_RDR-halfword read access

63.8.7.7.1 TDRE and RDRF Configuration

The TDRE flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_SPI_FMR.TXRDYM/RXRDYM.

As an example, if two data are written each time in FLEX_SPI_TDR, the TXRDYM field can be configured so that the TDRE flag is at ‘1’ only when at least two data can be written in the Transmit FIFO.

Similarly, if four data are read each time in FLEX_SPI_RDR, the RXRDYM field can be configured so that the RDRF flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

63.8.7.7.2 DMAC

It is mandatory to configure DMAC channel size (byte, halfword or word) according to the FLEX_SPI_FMR.TXRDYM/RXRDYM configuration. See constraints in SPI Multiple Data Access.

As an example, when FIFO is enabled, FLEX_SPI_FMR.TXRDYM/RXRDYM=0 configuration is not compatible with DMAC_PDC transfers in word (32-bit).

63.8.7.8 FIFO Pointer Error

A FIFO overflow is reported in FLEX_SPI_SR.

If the Transmit FIFO is full and a write access is performed on FLEX_SPI_TDR, it generates a Transmit FIFO pointer error and sets FLEX_SPI_SR.TXFPTEF.

If the number of data written in FLEX_SPI_TDR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_SPI_SR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_SPI_SR.

If the number of data read in FLEX_SPI_RDR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_SPI_SR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_SPI_TDR/SPI_RDR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a pointer error occurs, a software reset must be performed using FLEX_SPI_CR.SWRST (configuration will be lost).

63.8.7.9 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_SPI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from ‘above threshold’ to ‘equal to or below threshold’, the flag FLEX_SPI_SR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_SPI_FMR.RXFTHRES. Each time the Receive FIFO level goes from ‘below threshold’ to ‘equal to or above threshold’, the flag FLEX_SPI_SR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_SPI_IER and FLEX_SPI_IDR.

63.8.7.10 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_SPI_IER and FLEX_SPI_IDR.

FIFO flags state can be read in FLEX_SPI_SR. They are cleared when FLEX_SPI_SR is read.

63.8.8 SPI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_SPI to enable access to the write protection registers.

To prevent any single software error from corrupting SPI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SPI Write Protection Mode Register (FLEX_SPI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the SPI Write Protection Status Register (FLEX_SPI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_SPI_WPSR.

The following registers can be write-protected when WPEN is set:

- SPI Mode Register
- SPI Chip Select Register
- SPI Comparison Register

The following registers can be write-protected when WPITEN is set:

- SPI Interrupt Enable Register
- SPI Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- SPI Control Register

63.8.9 Local Loopback Test Mode

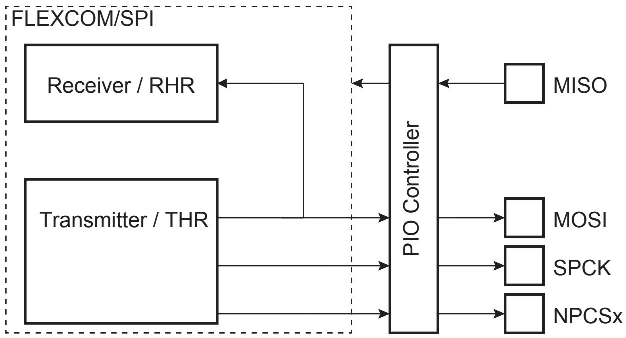
Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the figure below. The MISO pin has no effect on the receiver and the MOSI pin is driven as in Normal mode.

The MOSI, SPCK and NPCSx are normally transmitted unless the PIO is configured to drive logical values to prevent the SPI external target device from starting.

Local Loopback mode and allows a quick and easy verification of the transmitter and receiver logic.

Local Loopback mode is enabled when FLEX_SPI_MR.LLB=1, FLEX_SPI_MR.MASTER=1 and the FLEX_MR.OPMODE=2.

Figure 63-83. Local Loopback Mode Configuration

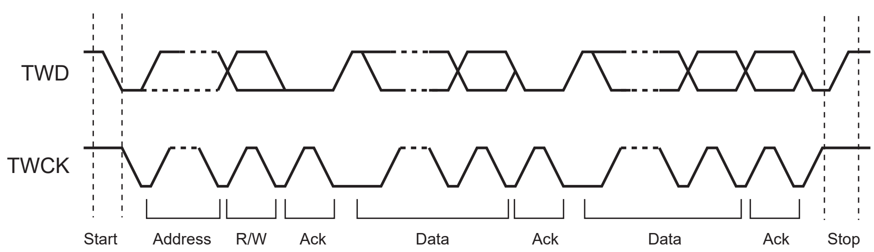


63.9 TWI Functional Description

63.9.1 Transfer Format

The data put on the TWD line must be 8 bits long. Data are transferred MSB first; each byte must be followed by an acknowledgement. The number of bytes per transfer is unlimited (see figure below).

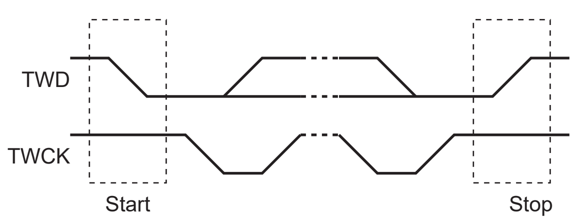
Figure 63-84. Transfer Format



Each transfer begins with a START condition and terminates with a STOP condition (see figure below).

- A high-to-low transition on the TWD line while TWCK is high defines the START condition.
- A low-to-high transition on the TWD line while TWCK is high defines a STOP condition.

Figure 63-85. START and STOP Conditions



63.9.1.1 Digital Filter

The TWI features digital filters on data and clock lines that can be configured by software via the TWI Filter register (FLEX_TWI_FILTR).

In Standard, Fast and Fast Plus modes, the digital filter must be enabled (FILT=1) and a pulse width threshold defined (THRES > 0).

The field THRES must be set according to the peripheral clock to suppress spikes below 50 ns. The recommended value is calculated using the formula:

$THRES > 50\ ns/t_{peripheral\_clock(ns)}$

63.9.2 Modes of Operation

The TWI has different modes of operation:

- Host Transmitter mode
- Host Receiver mode
- Multi-host Transmitter mode
- Multi-host Receiver mode
- Client Transmitter mode
- Client Receiver mode

These modes are described in the following sections.

63.9.3 Host Mode

63.9.3.1 Definition

The host is the device that starts a transfer, generates a clock and stops it.

63.9.3.2 Programming Host Mode

The following fields must be programmed before entering Host mode:

1. DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access client devices in Read or Write mode.
2. CWGR + CKDIV + CHDIV + CLDIV: Clock waveform.
3. SVDIS: Disables Client mode.
4. MSEN: Enables Host mode.

Note: If the TWI is already in Host mode, the device address (DADR) can be configured without disabling Host mode.

63.9.3.3 Transfer Speed/Bit Rate

The TWI speed is defined in FLEX_TWI_CWGR. The TWI bit rate can be based either on the peripheral clock if the BRSRCCLK bit value is 0 or on a programmable clock source provided by the GCLK if the BRSRCCLK bit value is 1.

If BRSRCCLK = 1, the bit rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the TWI transfer rate.

The GCLK frequency must be at least three times lower than the peripheral clock frequency.

63.9.3.4 Host Transmitter Mode

After the host initiates a START condition when writing into the Transmit Holding register FLEX_TWI_THR, it sends a 7-bit client address, configured in the Host Mode register (DADR in FLEX_TWI_MMR), to notify the client device. The bit following the client address indicates the transfer direction, 0 in this case (FLEX_TWI_MMR.MREAD = 0).

The TWI transfers require the client to acknowledge each received byte. During the acknowledge clock pulse (ninth pulse), the host releases the data line (HIGH), enabling the client to pull it down in order to generate the acknowledge. If the client does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWI Status register (FLEX_TWI_SR) of the host and a STOP condition is sent. Alternatively, if the FLEX_TWI_MMR.NOAP bit is set, no stop condition will be sent and a START or STOP condition must be triggered manually through the FLEX_TWI_CR.START or FLEX_TWI_CR.STOP bit once the software is ready for the transmission of the condition. The NACK flag must be cleared by reading the TWI Status register (FLEX_TWI_SR) before the next write into the TWI Transmit Holding register (FLEX_TWI_THR). As with the other status bits, an interrupt can be generated if enabled in the Interrupt Enable register (FLEX_TWI_IER). If the client acknowledges the byte, the data written in FLEX_TWI_THR is then shifted in the internal shifter and transferred. When an acknowledge is detected, the TXRDY bit is set until a new write in FLEX_TWI_THR.

TXRDY is used as transmit ready for the DMA transmit channel.

Note: To clear the TXRDY flag in Host mode, write the FLEX_TWI_CR.MSDIS bit to 1, then write the FLEX_TWI_CR.MSEN bit to 1.

While no new data is written in FLEX_TWI_THR, the serial clock line is tied low. When new data is written in FLEX_TWI_THR, the SCL is released and the data is sent. To generate a STOP event, the STOP command must be performed by writing in the STOP field of the TWI Control register (FLEX_TWI_CR).

After a host write transfer, the Serial Clock line is stretched (tied low) while no new data is written in FLEX_TWI_THR or until a STOP command is performed.

See the following figures.

Figure 63-86. Host Write with One Data Byte

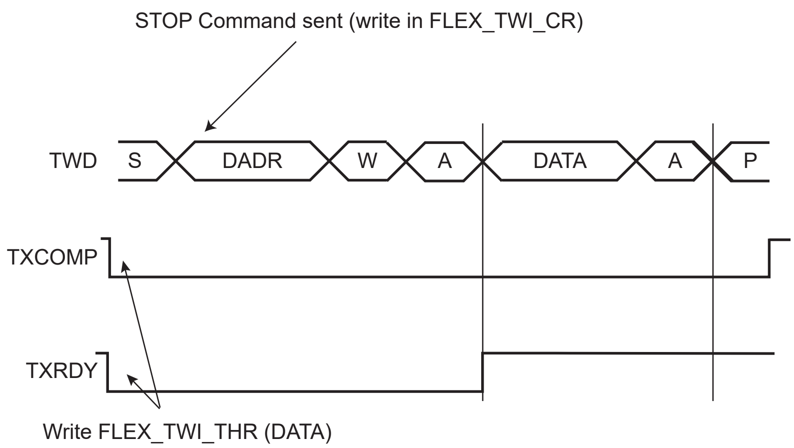


Figure 63-87. Host Write with Multiple Data Bytes

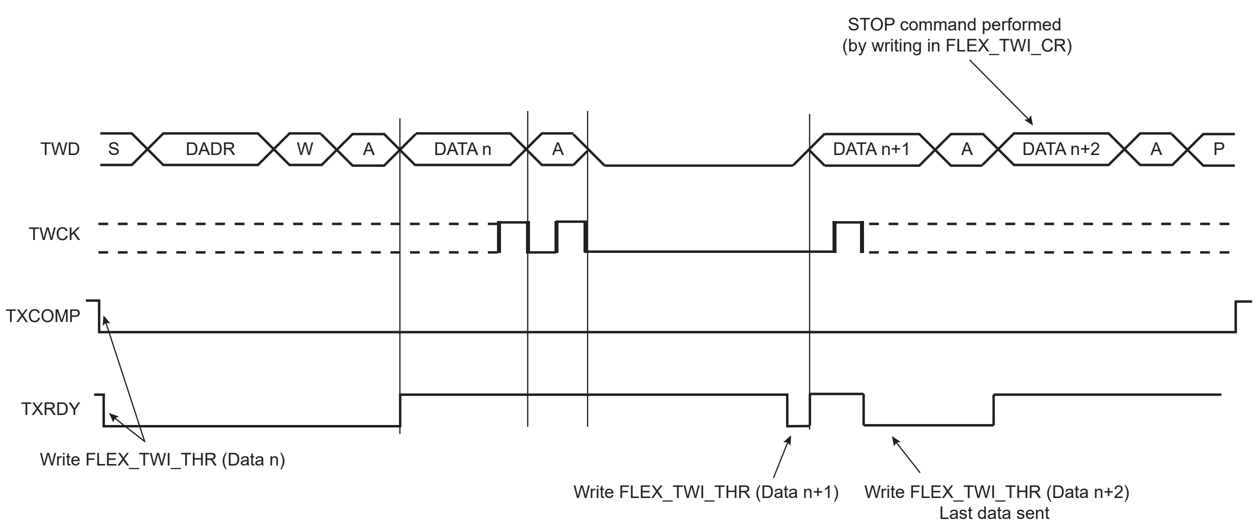
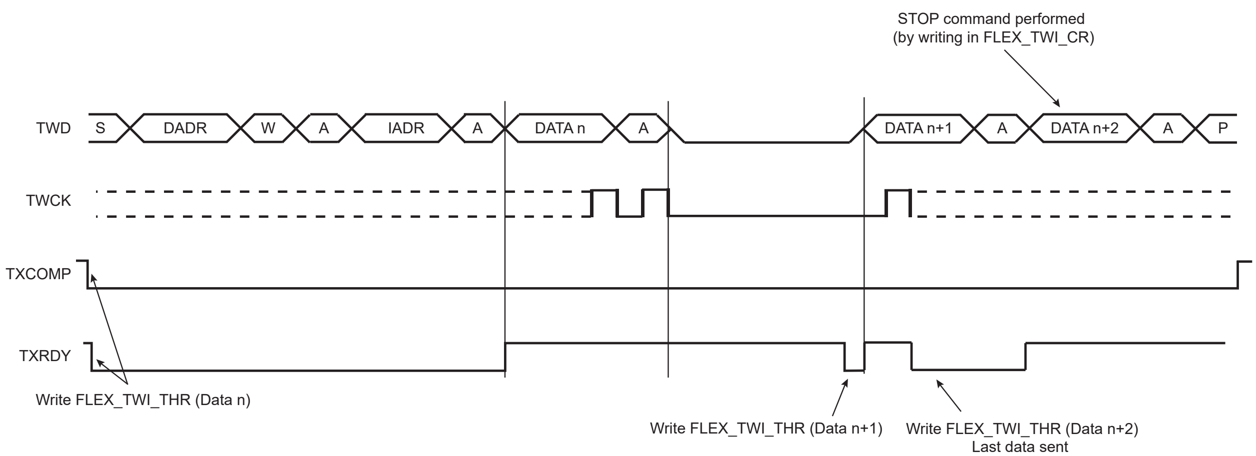


Figure 63-88. Host Write with One Byte Internal Address and Multiple Data Bytes



63.9.3.5 Host Receiver Mode

The read sequence begins by setting the START bit. After the start condition has been sent, the host sends a 7-bit client address to notify the client device. The bit following the client address indicates the transfer direction, 1 in this case (FLEX_TWI_MMR.MREAD = 1). During the acknowledge clock pulse (9th pulse), the host releases the data line (HIGH), enabling the client to pull it down in order to generate the acknowledge. The host polls the data line during this clock pulse and sets the FLEX_TWI_SR.NACK bit if the client does not acknowledge the byte.

If an acknowledge is received, the host is then ready to receive data from the client. After data has been received, the host sends an acknowledge condition to notify the client that the data has been received except for the last data (see figure "Host Read with One Data Byte" below). When the FLEX_TWI_SR.RXRDY bit is set, a character has been received in the Receive Holding register (FLEX_TWI_RHR). The RXRDY bit is reset when reading FLEX_TWI_RHR.

When a single data byte read is performed, with or without internal address (IADR), the START and STOP bits must be set at the same time. See figure "Host Read with One Data Byte" below. When a multiple data byte read is performed, with or without internal address (IADR), the STOP bit must be set after the next-to-last data received (same condition applies for START bit to generate a repeated start). See figure "Host Read with Multiple Data Bytes" below. For internal address usage, see Internal Address.

If FLEX_TWI_RHR is full (RXRDY high) and the host is receiving data, the serial clock line will be tied low before receiving the last bit of the data and until FLEX_TWI_RHR is read. Once FLEX_TWI_RHR is read, the host will stop stretching the serial clock line and end the data reception. See figure "Host Read Clock Stretching with Multiple Data Bytes" below.

⚠WARNING	When receiving multiple bytes in Host Read mode, if the next-to-last access is not read (the RXRDY flag remains high), the last access will not be completed until FLEX_TWI_RHR is read. The last access stops on the next-to-last bit (clock stretching). When FLEX_TWI_RHR is read there is only half a bit period to send the STOP bit (or START bit) command, else another read access might occur (spurious access). A possible workaround is to set the STOP bit (or START bit) before reading FLEX_TWI_RHR on the next-to-last access (within IT handler).

Figure 63-89. Host Read with One Data Byte

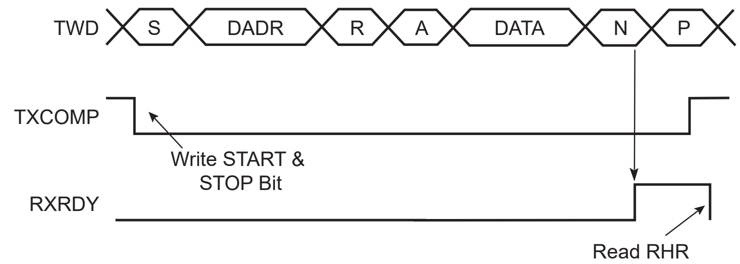


Figure 63-90. Host Read with Multiple Data Bytes

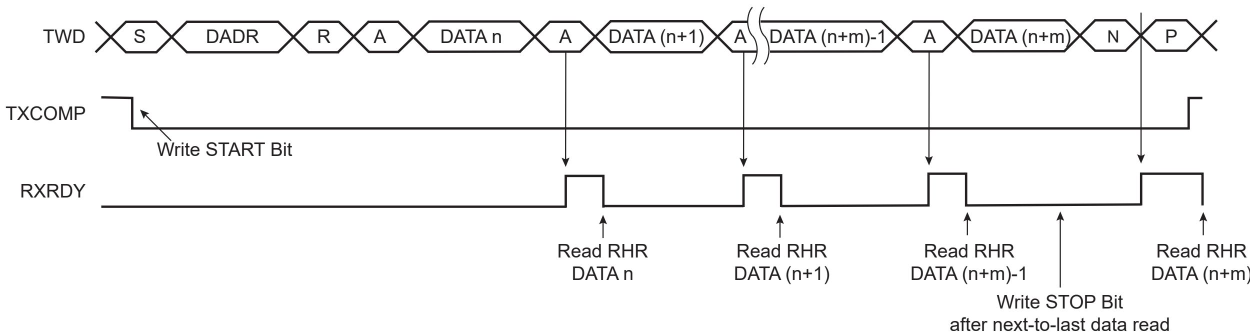
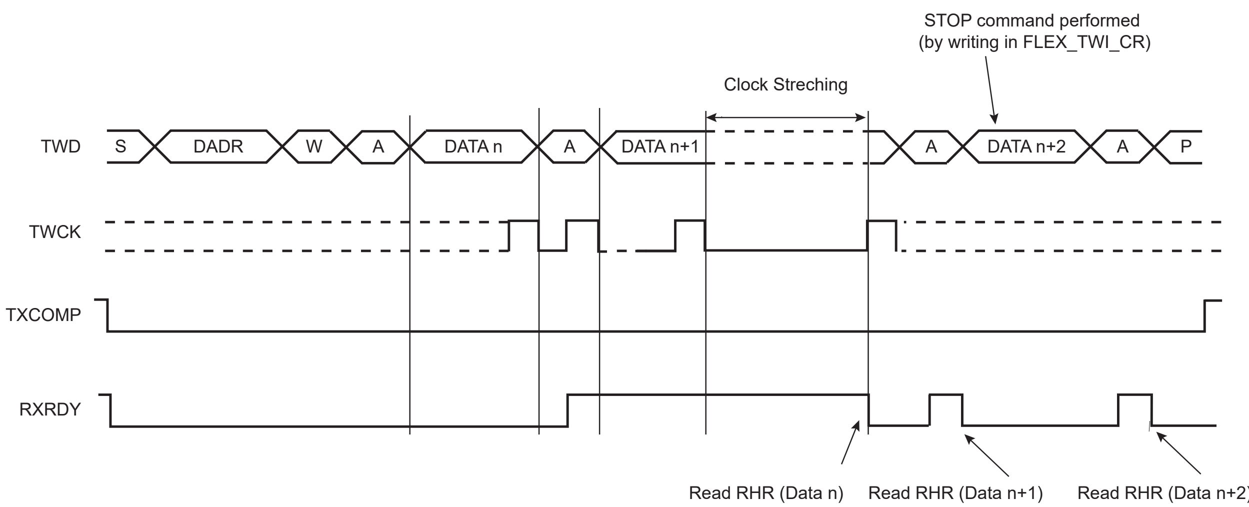


Figure 63-91. Host Read Clock Stretching with Multiple Data Bytes



RXRDY is used as receive ready trigger event for the DMA receive channel.

63.9.3.6 Internal Address

The TWI interface can perform transfers with 7-bit client address devices and with 10-bit client address devices.

63.9.3.6.1 7-bit Client Addressing

When addressing 7-bit client devices, the internal address bytes are used to perform random address (read or write) accesses to reach one or more data bytes, e.g., within a memory page location in a serial memory. When performing read operations with an internal address, the TWI performs a write operation to set the internal address into the client device, and then switch to Host Receiver mode. Note that the second start condition (after sending the IADR) is sometimes called “repeated start” (Sr) in I2C fully-compatible devices. See figure Host Read with One, Two or Three Bytes Internal Address and One Data Byte.

See figures Host Write with One, Two or Three Bytes Internal Address and One Data Byte and Internal Address Usage for the host write operation with internal address.

The three internal address bytes are configurable through the Host Mode register (FLEX_TWI_MMR).

If the client device supports only a 7-bit address, that is, no internal address, IADRSZ must be configured to 0.

The abbreviations listed below are used in the following figures:

S	Start
Sr	Repeated Start
P	Stop
W	Write
R	Read
A	Acknowledge
N	Not Acknowledge
DADR	Device Address
IADR	Internal Address

Figure 63-92. Host Write with One, Two or Three Bytes Internal Address and One Data Byte

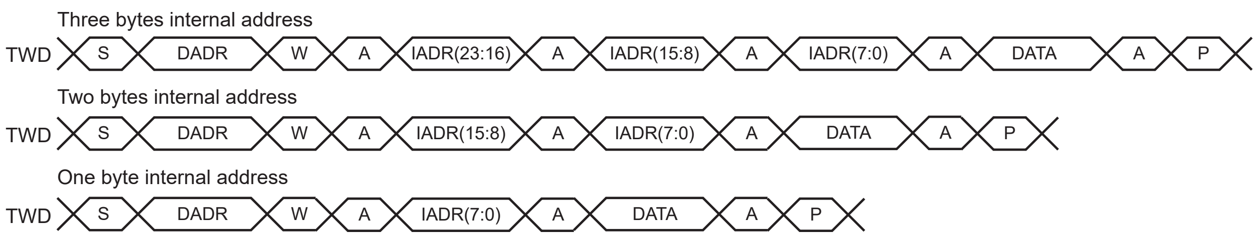
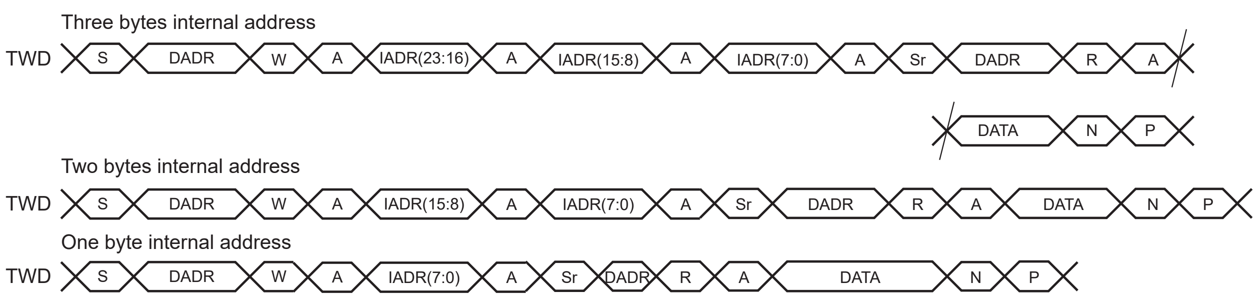


Figure 63-93. Host Read with One, Two or Three Bytes Internal Address and One Data Byte



63.9.3.6.2 10-bit Client Addressing

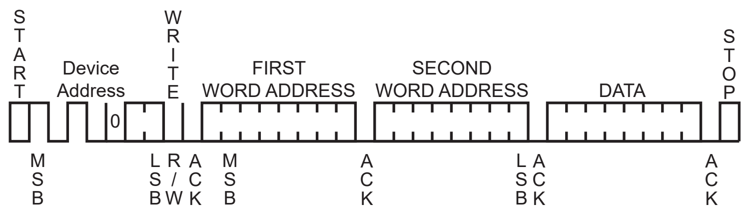
For a client address higher than seven bits, the user must configure the address size (IADRSZ) and set the other client address bits in the Internal Address register (FLEX_TWI_IADR). The two remaining internal address bytes, IADR[15:8] and IADR[23:16], can be used the same way as in 7-bit client addressing.

Example: Address a 10-bit device (10-bit device address is b1 b2 b3 b4 b5 b6 b7 b8 b9 b10)

1. Program IADRSZ = 1.
2. Program DADR with 1 1 1 1 0 b1 b2 (b1 is the MSB of the 10-bit address, b2, etc.).
3. Program FLEX_TWI_IADR with b3 b4 b5 b6 b7 b8 b9 b10 (b10 is the LSB of the 10-bit address).

The following figure shows a byte write to a TWI EEPROM. This demonstrates the use of internal addresses to access the device.

Figure 63-94. Internal Address Usage



63.9.3.7 Repeated Start

In addition to Internal Address mode, repeated start (Sr) can be generated manually by writing the START bit at the end of a transfer instead of the STOP bit. In such case the parameters of the next transfer (direction, SADR, etc.) will need to be set before writing the START bit at the end of the previous transfer.

See Read/Write Flowcharts.

63.9.3.8 Bus Clear Command

The TWI interface can perform a Bus Clear Command:

1. Configure Host mode (DADR, CKDIV, etc).
2. Start the transfer by setting the FLEX_TWI_CR.CLEAR bit.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

63.9.3.9 SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMBEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.
2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses has been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

63.9.3.9.1 Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one enables automatic PEC handling in the current transfer. Transfers with and without PEC can freely be intermixed in the same system, since some clients may not support PEC. The PEC LFSR is always updated on every bit transmitted or received, so that PEC handling on combined transfers will be correct.

In Host Transmitter mode, the host calculates a PEC value and transmits it to the client after all data bytes have been transmitted. Upon reception of this PEC byte, the client will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the client will return an ACK to the host. If the PEC values differ, data was corrupted, and the client will return a NACK value. Some clients may not be able to check the received PEC in time to return a NACK if an error occurred. In this case, the client should always return an ACK after the PEC byte, and some other mechanism must be implemented to verify that the transmission was received correctly.

In Host Receiver mode, the client calculates a PEC value and transmits it to the host after all data bytes have been transmitted. Upon reception of this PEC byte, the host will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the FLEX_TWI_SR.PECERR bit is set. In Host Receiver mode, the PEC byte is always followed by a NACK transmitted by the host, since it is the last byte in the transfer.

In combined transfers, the PECRQ bit should only be set in the last of the combined transfers. If Alternative Command mode is enabled, only the NPEC bit should be set.

Consider the following transfer:

S, ADR+W, COMMAND_BYTE, ACK, SR, ADR+R, DATA_BYTE, ACK, PEC_BYTE, NACK, P

See Read/Write Flowcharts for detailed flowcharts.

63.9.3.9.2 Timeouts

The FLEX_TWI_SMBTR.TLOWS/TLOWM fields configure the SMBus timeout values. If a timeout occurs, the host transmits a STOP condition and leaves the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

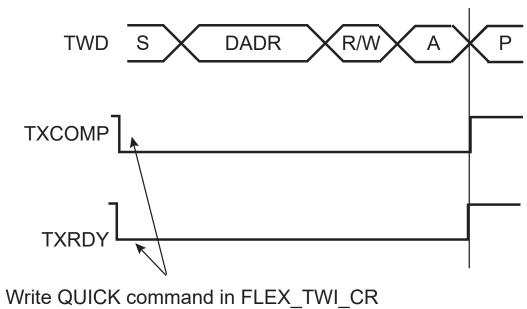
63.9.3.10 SMBus Quick Command (Host Mode Only)

The TWI interface can perform a quick command:

1. Configure Host mode (DADR, CKDIV, etc).
2. Write the FLEX_TWI_MMR.MREAD bit at the value of the one-bit command to be sent.
3. Start the transfer by setting the FLEX_TWI_CR.QUICK bit.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

Figure 63-95. SMBus Quick Command



63.9.3.11 TWI High-Speed Host

TWI High-Speed Host mode is enabled when FLEX_TWI_CR.HSEN=1 and FLEX_TWI_CR.MSEN=1. TWI High-Speed mode operation is similar to TWI operation, with the following exceptions:

1. A host code is sent first at normal speed before enabling TWI High-Speed mode.
2. Arbitration is only possible during host code transmission.
3. When TWI High-Speed mode is active, an internal mechanism is enabled to shorten the SCL signal rise time
4. When TWI High-Speed mode is active, clock stretching is only allowed after acknowledge (ACK), not-acknowledge (NACK), START (S) or repeated START (Sr) (as a consequence, overflow can happen).
5. When TWI High-Speed mode is active, the data transfer uses the TWI High Speed mode bit rate which can be defined in FLEX_TWI_HSCWGR.

TWI High-Speed mode allows transfers up to 3.4 Mbits/s.

63.9.3.11.1 Read-Write Operation

The TWI high-speed frame always begins with the following sequence:

1. START condition (S)
2. Host code (0000 1XXX)
3. Not-acknowledge (NACK)

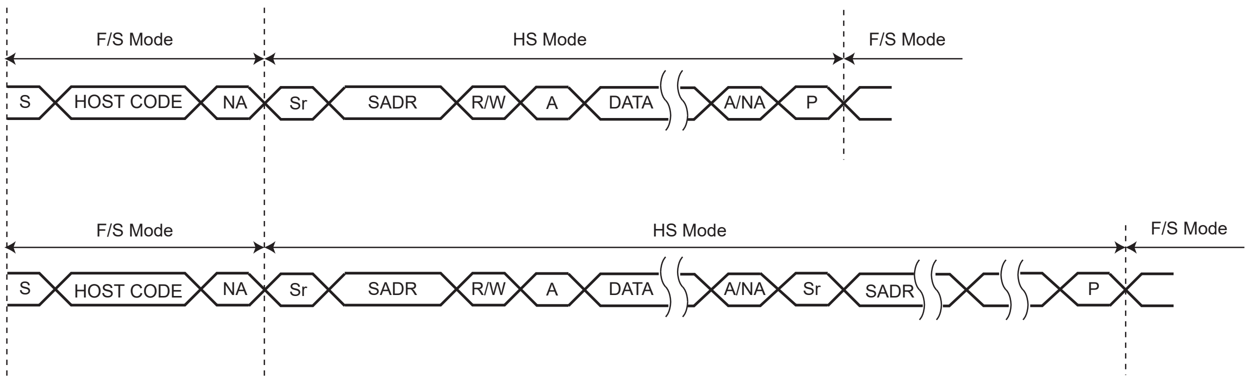
Note that to enable high-speed transfer, the client must send a NACK after receiving the host code.

As described previously, this sequence is sent at normal speed, and in case of multi-host, arbitration can only happen during this sequence.

Host codes are 8-bit reserved codes with the sole purpose of enabling High-Speed mode. Each host must use a different host code, which can be set through the FLEX_TWI_HSR.MCODE field.

Once this sequence has been sent, High-Speed mode is enabled and the host sends a repeated start to begin the programmed transfer. High-Speed mode remains enabled after repeated START (Sr) and only switches back to normal speed after a STOP condition (P).

Figure 63-96. High-Speed Mode Read-Write



63.9.3.11.2 SCL Rising Time Control

In order to meet the TWI High-Speed mode SCL rise time requirements, the SCL Rise Boost feature is enabled automatically when TWI High-Speed mode is enabled.

SCLRBL bit in FLEX_TWI_MMR enables to set the number of system clock periods (MCK) during which the SCL signal will be directly driven to level ‘1’ when the SCL rising edge is sent. This short time during which the SCL pin is directly driven to level ‘1’ allows to increase the SCL slope as much as needed to meet the High-Speed mode rise time requirements.

63.9.3.11.3 TWI High-Speed Mode Usage

TWI High-Speed mode usage is the same as for standard TWI (see Read/Write Flowcharts).

63.9.3.12 Alternative Command

Another way to configure the transfer is to enable the Alternative Command mode with the ACMEN bit of the TWI Control Register.

In this mode, the transfer is configured through the TWI Alternative Command Register. It is possible to define a simple read or write transfer or a combined transfer with a repeated start.

In order to set a simple transfer, the DATAL field and the DIR field of the TWI Alternative Command Register must be filled accordingly and the NDATAL field must be cleared. To begin the transfer, either set the START bit in the TWI Control Register in case of a read transfer, or write the TWI Transmit Holding Register in case of a write transfer.

For a combined transfer linked by a repeated start, the NDATAL field must be filled with the length of the second transfer and NDIR with the corresponding direction.

The PEC and NPEC bits are used to set a PEC field. In the case of a single transfer with PEC, the PEC bit must be set. In the case of a combined transfer, the NPEC bit must be set.

Note: If the Alternative Command mode is used, the TWIHS_MMR.IADRSZ field must be set to 0.

See Read/Write Flowcharts for detailed flowcharts.

63.9.3.13 Handling Errors in Alternative Command

In case of NACK generated by a client device or SMBus timeout error, the TWI stops immediately the frame, but the DMA transfer may still be active. To prevent a new frame to be restarted with the remaining DMA data (transmit), the TWI prevents any start of frame until the FLEX_TWI_SR.LOCK flag is cleared.

The FLEX_TWI_SR.LOCK bit indicates the state of the TWI (locked or not locked).

When the TWI is locked, no transfer can begin until the LOCK is cleared using the FLEX_TWI_CR.LOCKCLR bit and until the error flags are cleared reading FLEX_TWI_SR.

In case of error, FLEX_TWI_THR may have been loaded with a new data. The FLEX_TWI_CR.THRCLR bit can be used to flush FLEX_TWI_THR. If the THRCLR bit is set, the TXRDY and TXCOMP flags are set.

63.9.3.14 Read/Write Flowcharts

The flowcharts shown in this section provide examples for read and write operations. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable register (FLEX_TWI_IER) be configured first.

Figure 63-97. TWI Write Operation with Single Data Byte without Internal Address

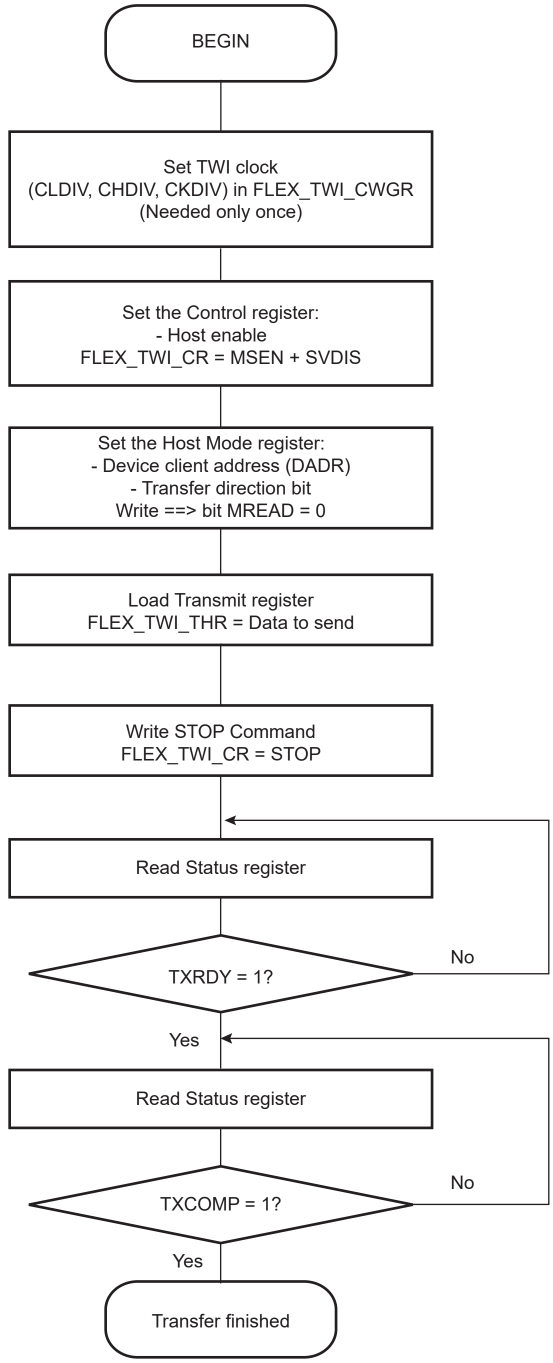


Figure 63-98. TWI Write Operation with Single Data Byte and Internal Address

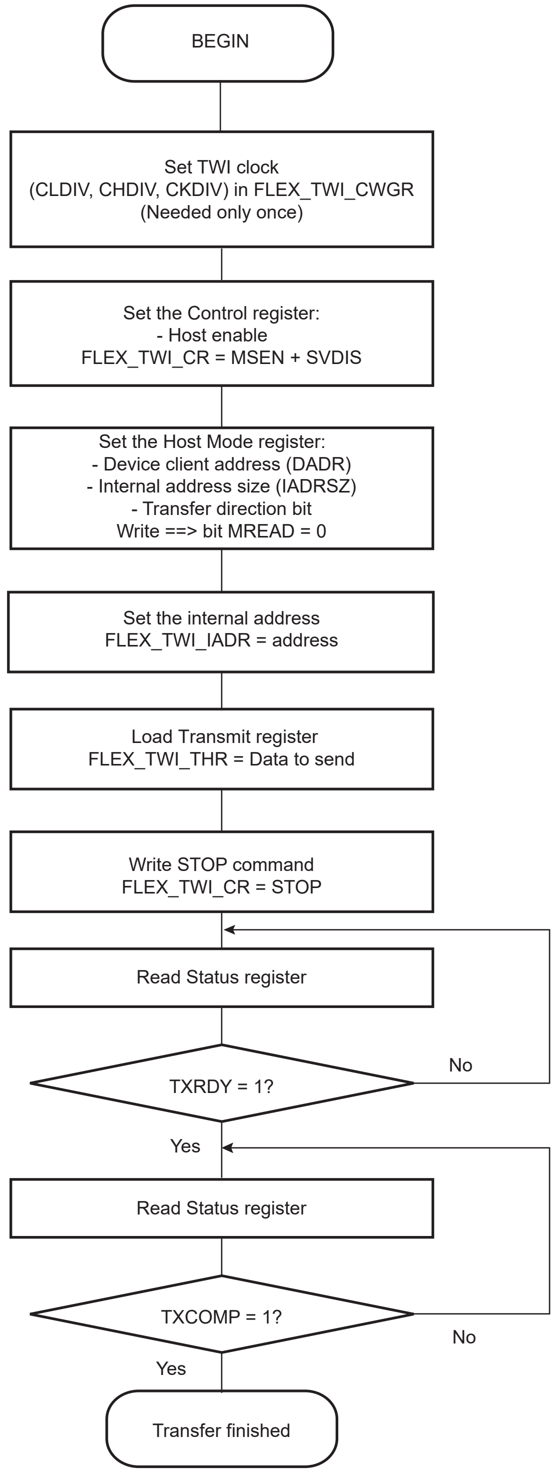


Figure 63-99. TWI Write Operation with Multiple Data Bytes with or without Internal Address

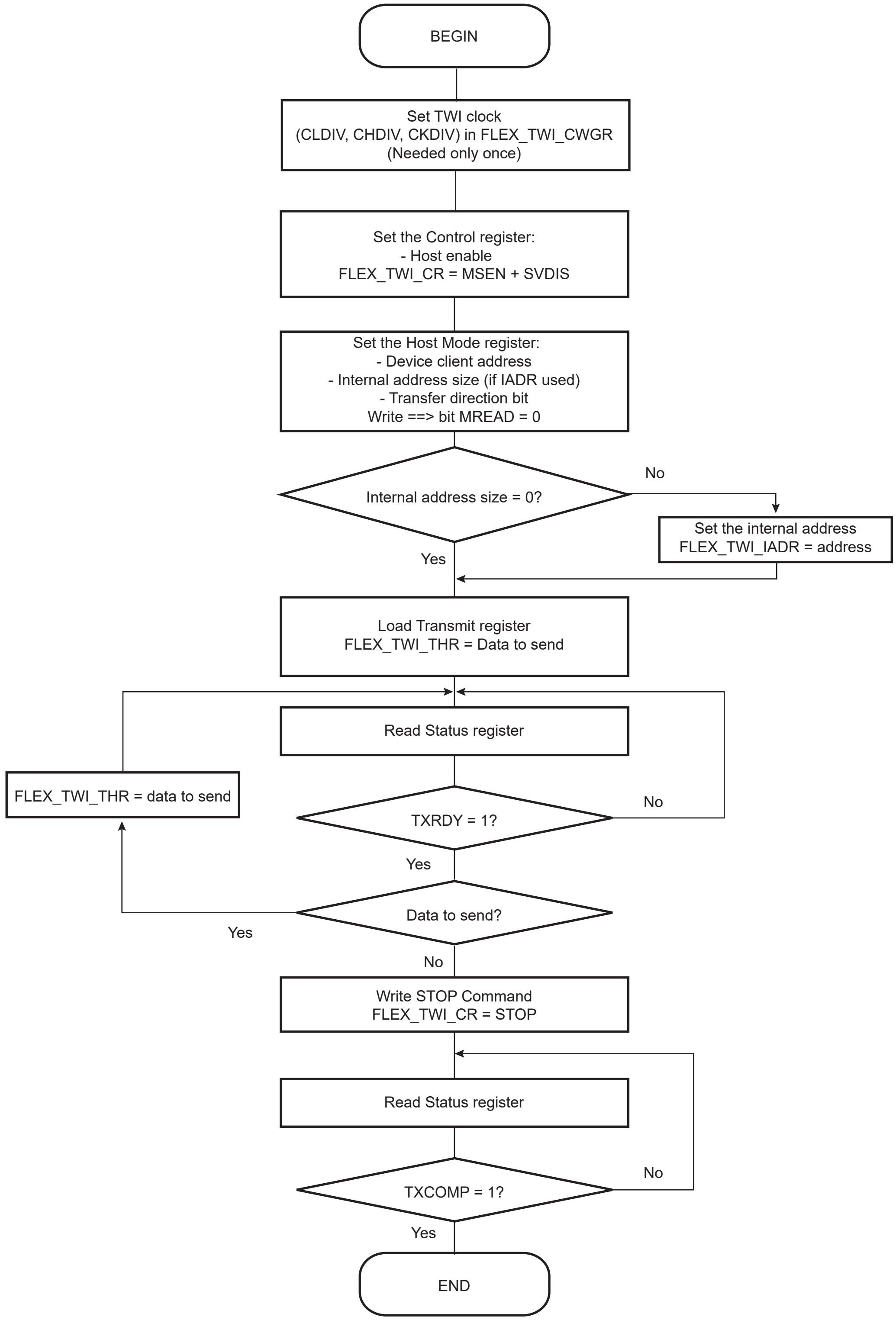


Figure 63-100. SMBus Write Operation with Multiple Data Bytes with or without Internal Address and PEC Sending

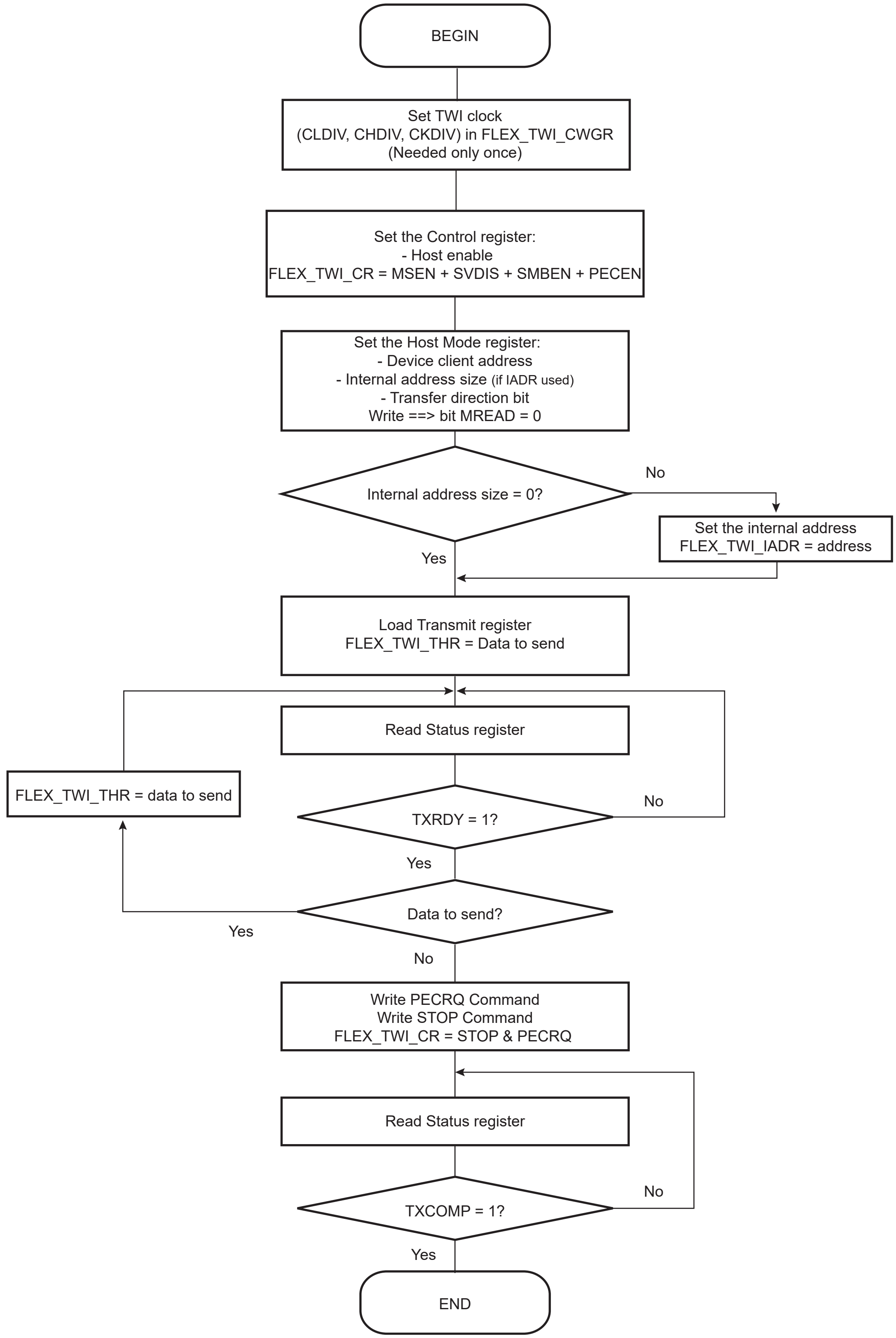


Figure 63-101. SMBus Write Operation with Multiple Data Bytes with PEC and Alternative Command Mode

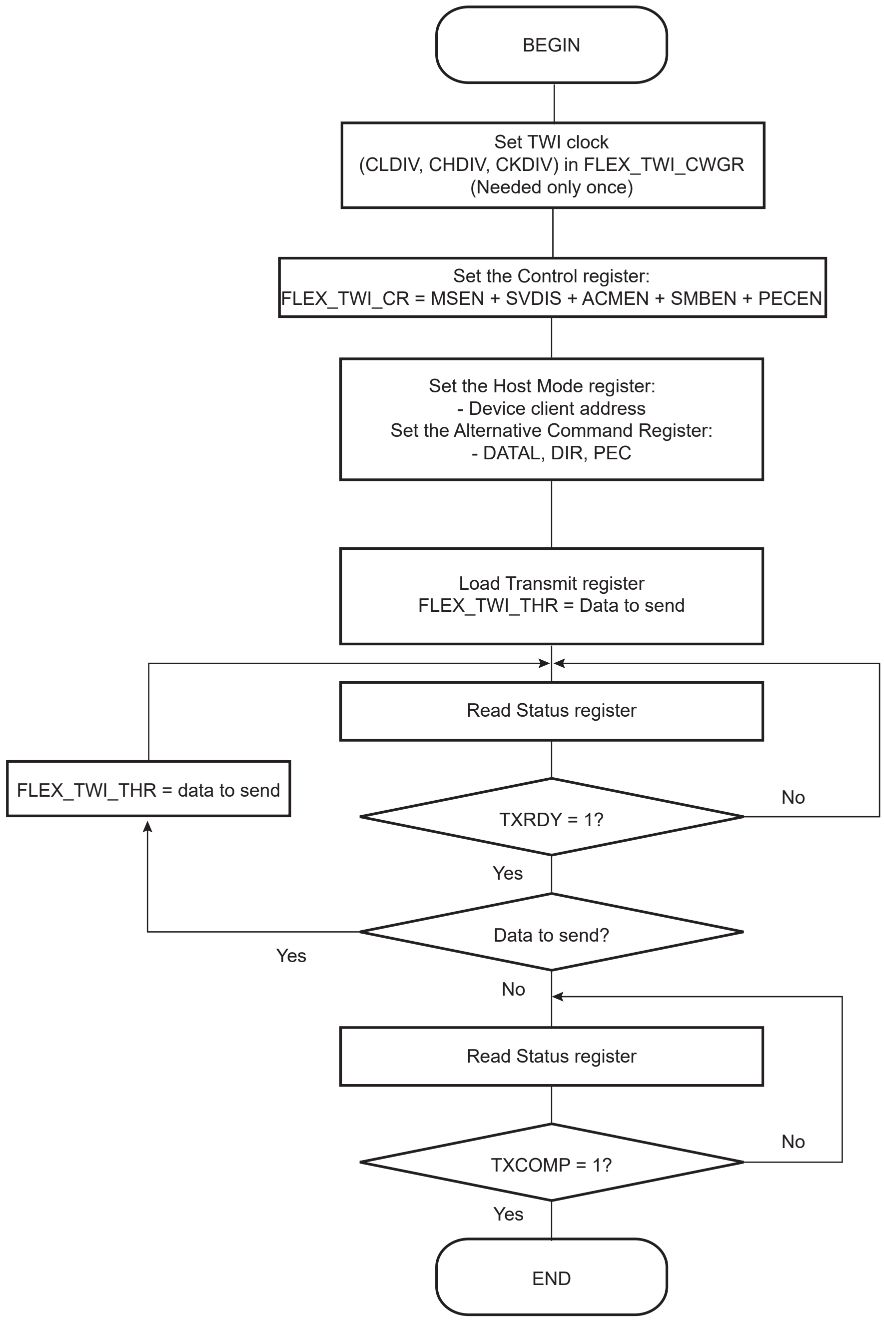


Figure 63-102. TWI Write Operation with Multiple Data Bytes and Read Operation with Multiple Data Bytes (Sr)

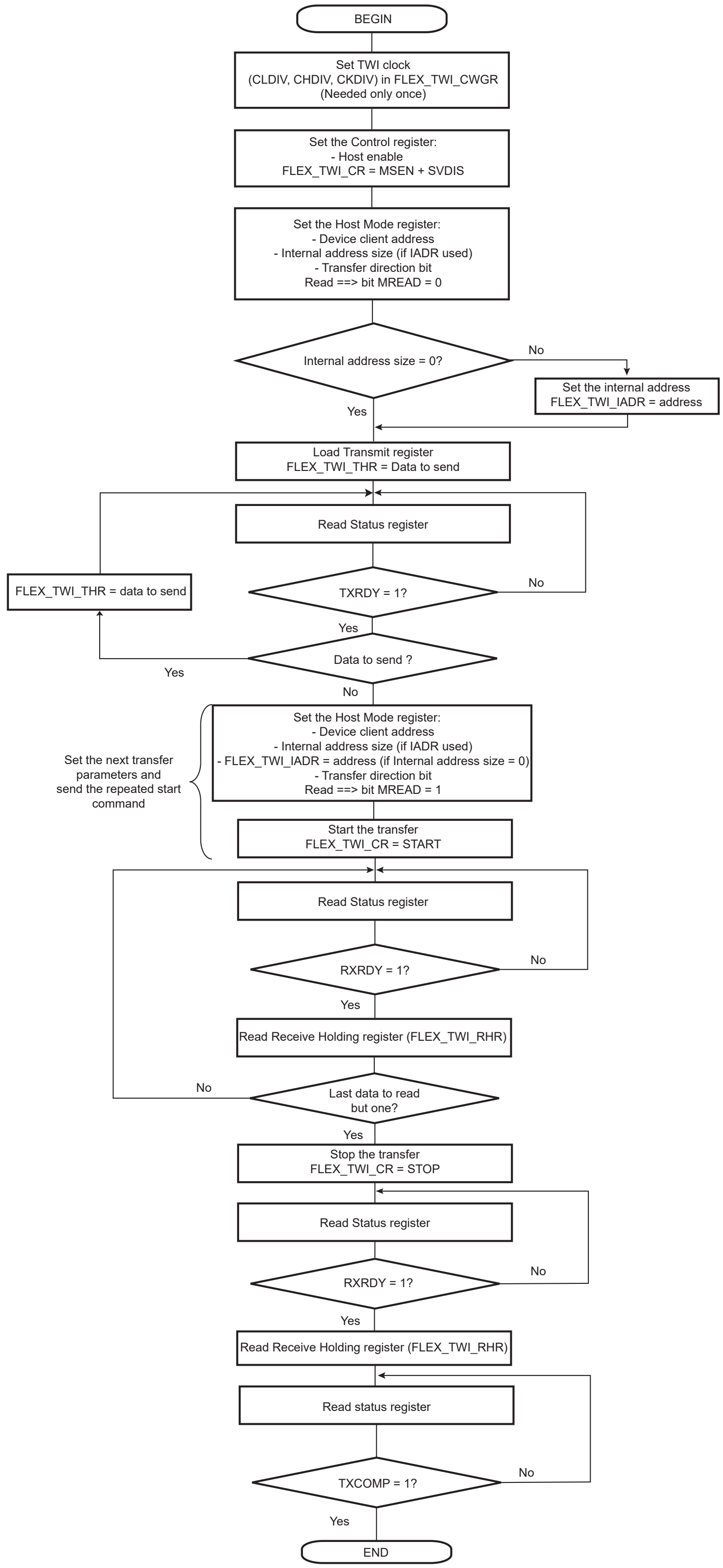


Figure 63-103. TWI Write Operation with Multiple Data Bytes + Read Operation and Alternative Command Mode + PEC

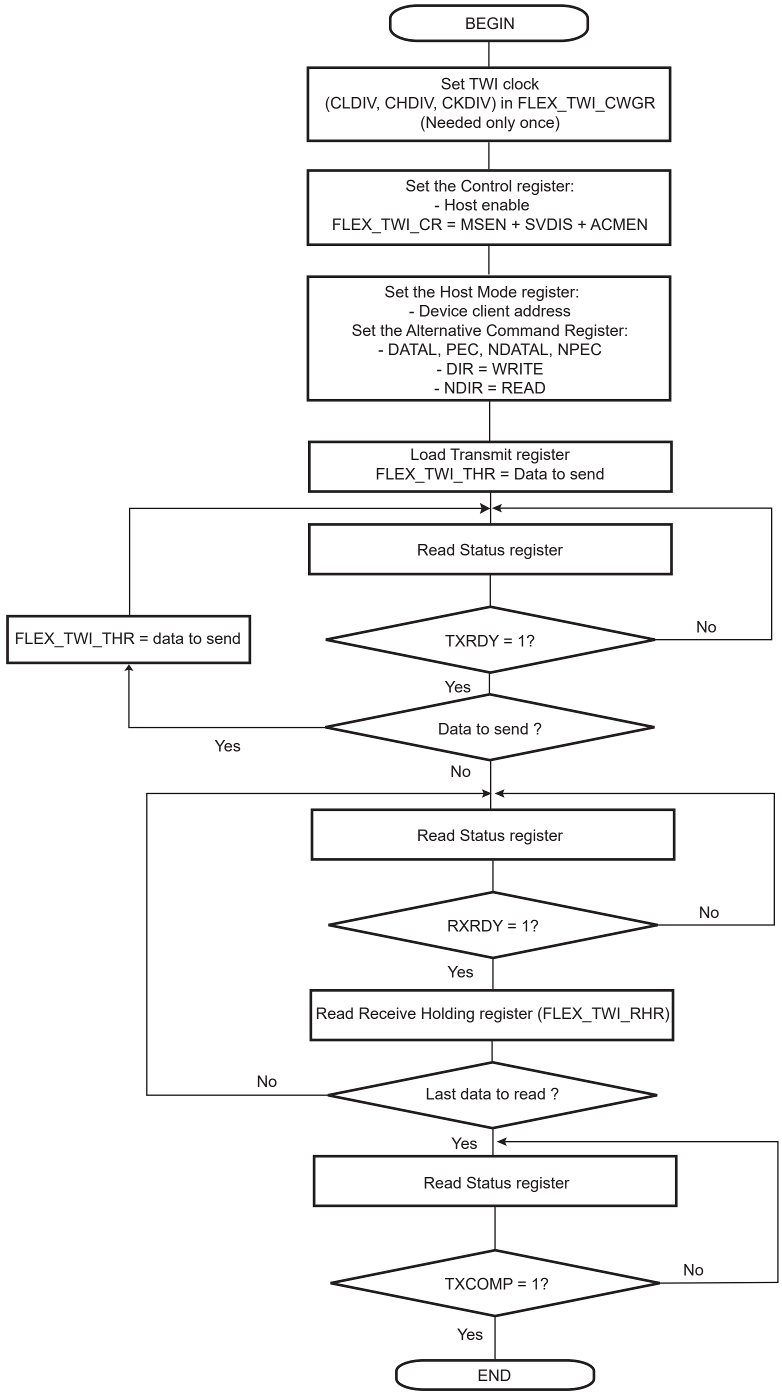


Figure 63-104. TWI Read Operation with Single Data Byte without Internal Address

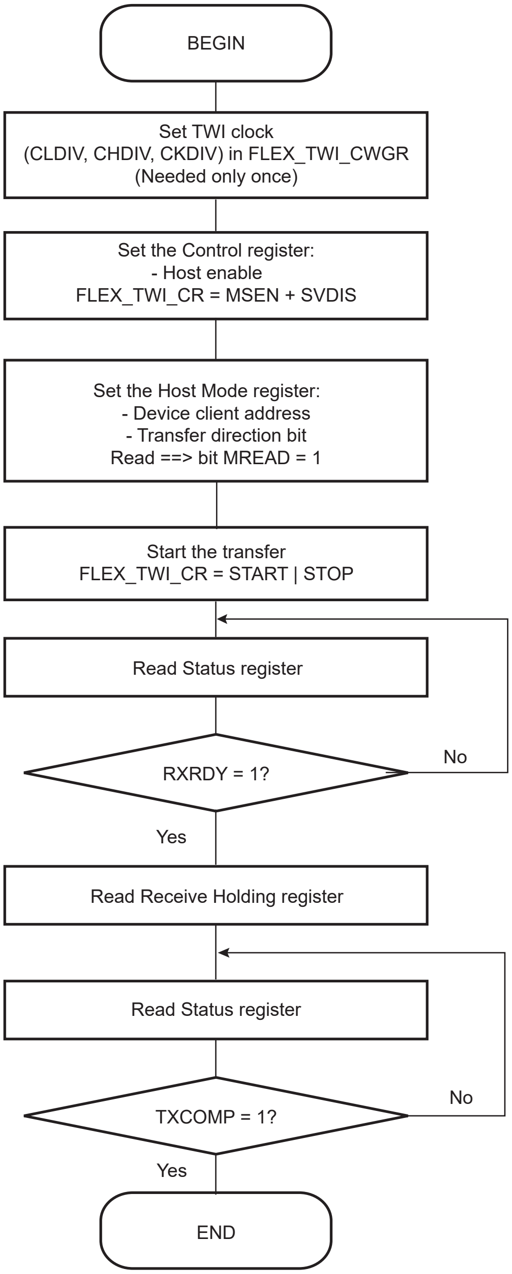


Figure 63-105. TWI Read Operation with Single Data Byte and Internal Address

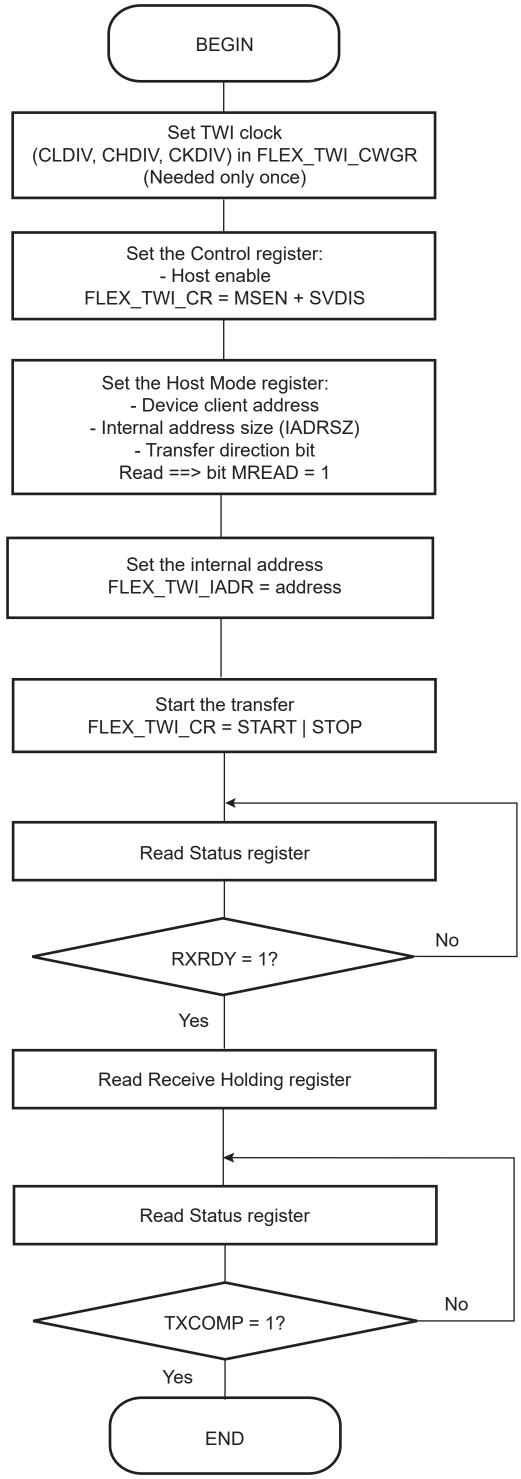


Figure 63-106. TWI Read Operation with Multiple Data Bytes with or without Internal Address

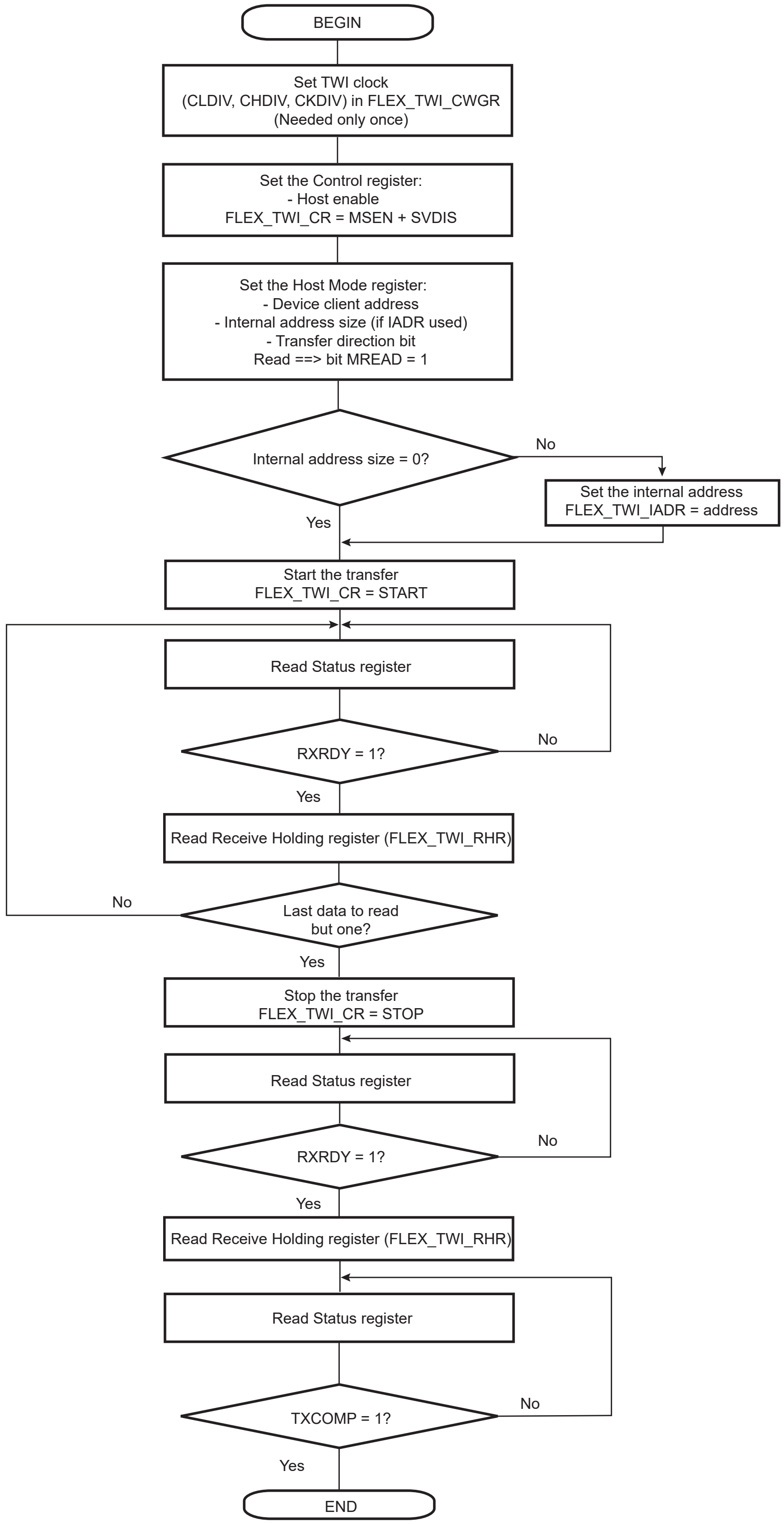


Figure 63-107. TWI Read Operation with Multiple Data Bytes with or without Internal Address with PEC

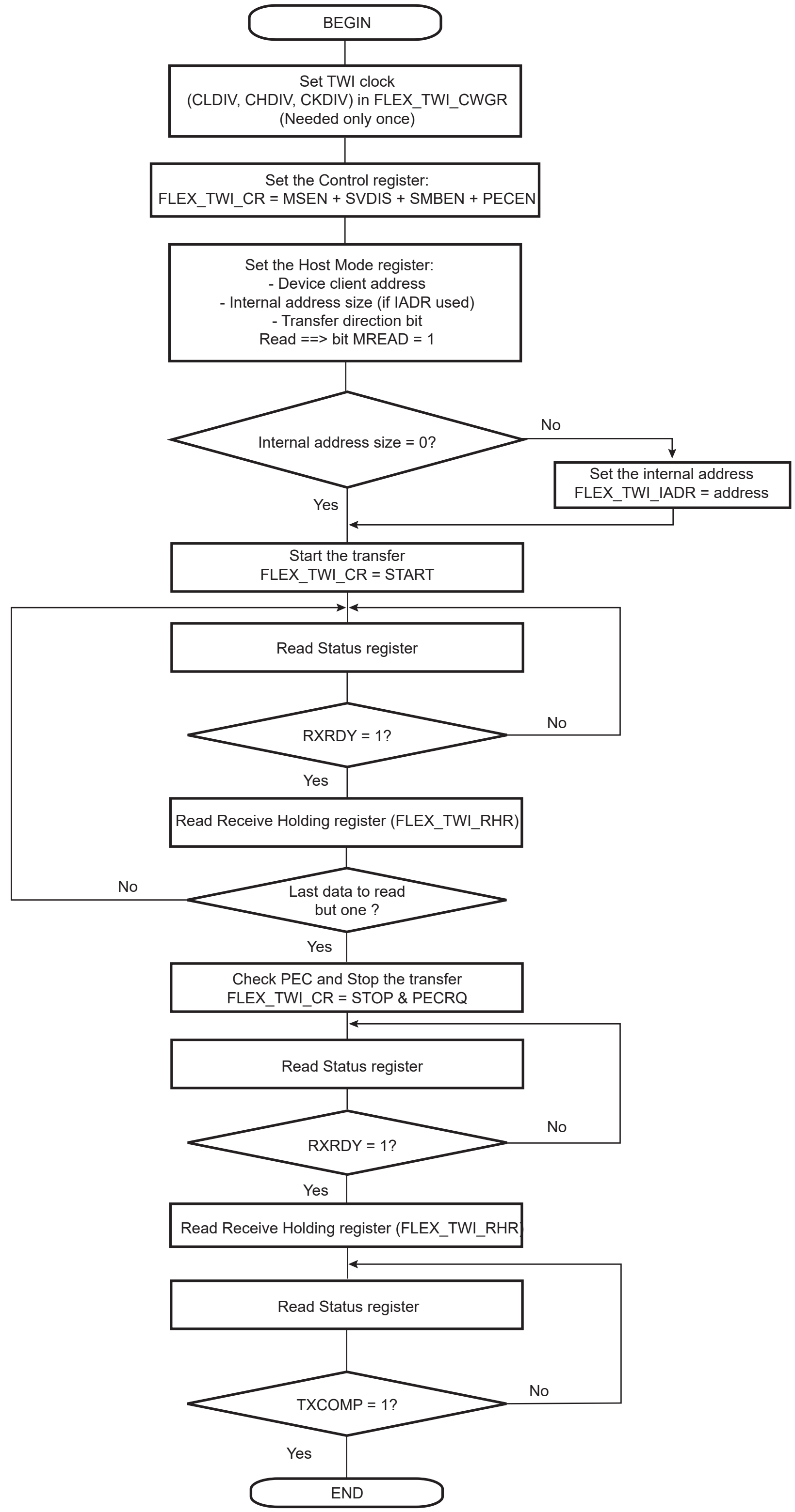


Figure 63-108. TWI Read Operation with Multiple Data Bytes with Alternative Command Mode with PEC

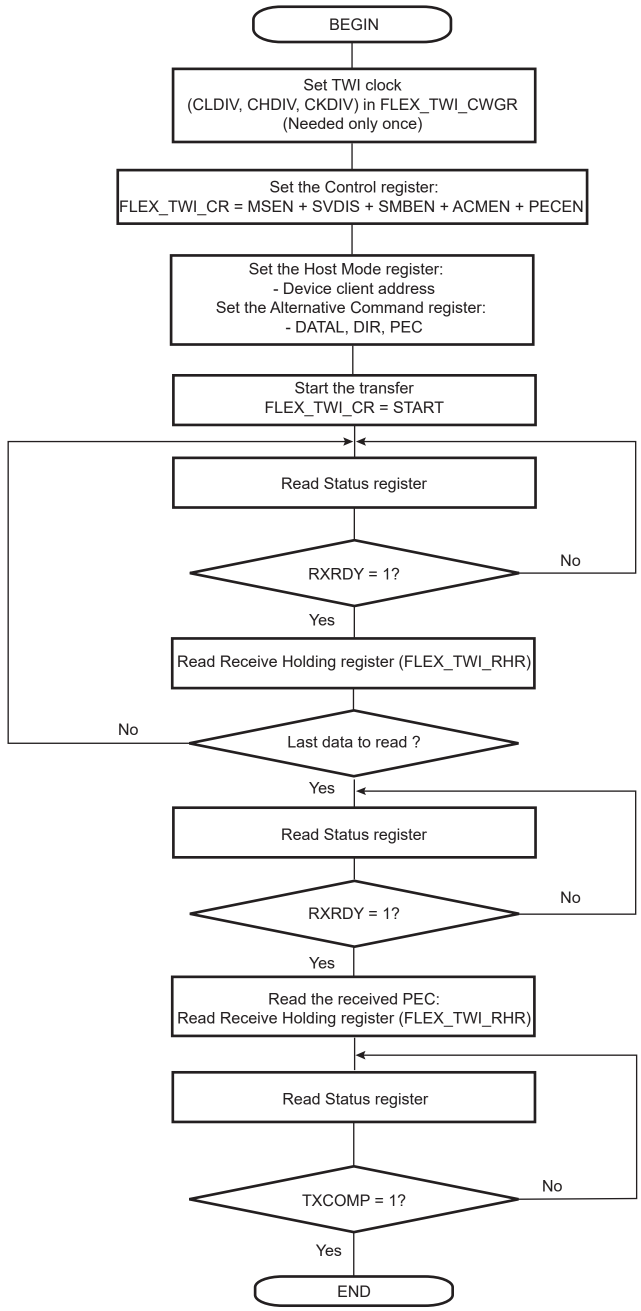


Figure 63-109. TWI Read Operation with Multiple Data Bytes + Write Operation with Multiple Data Bytes (Sr)

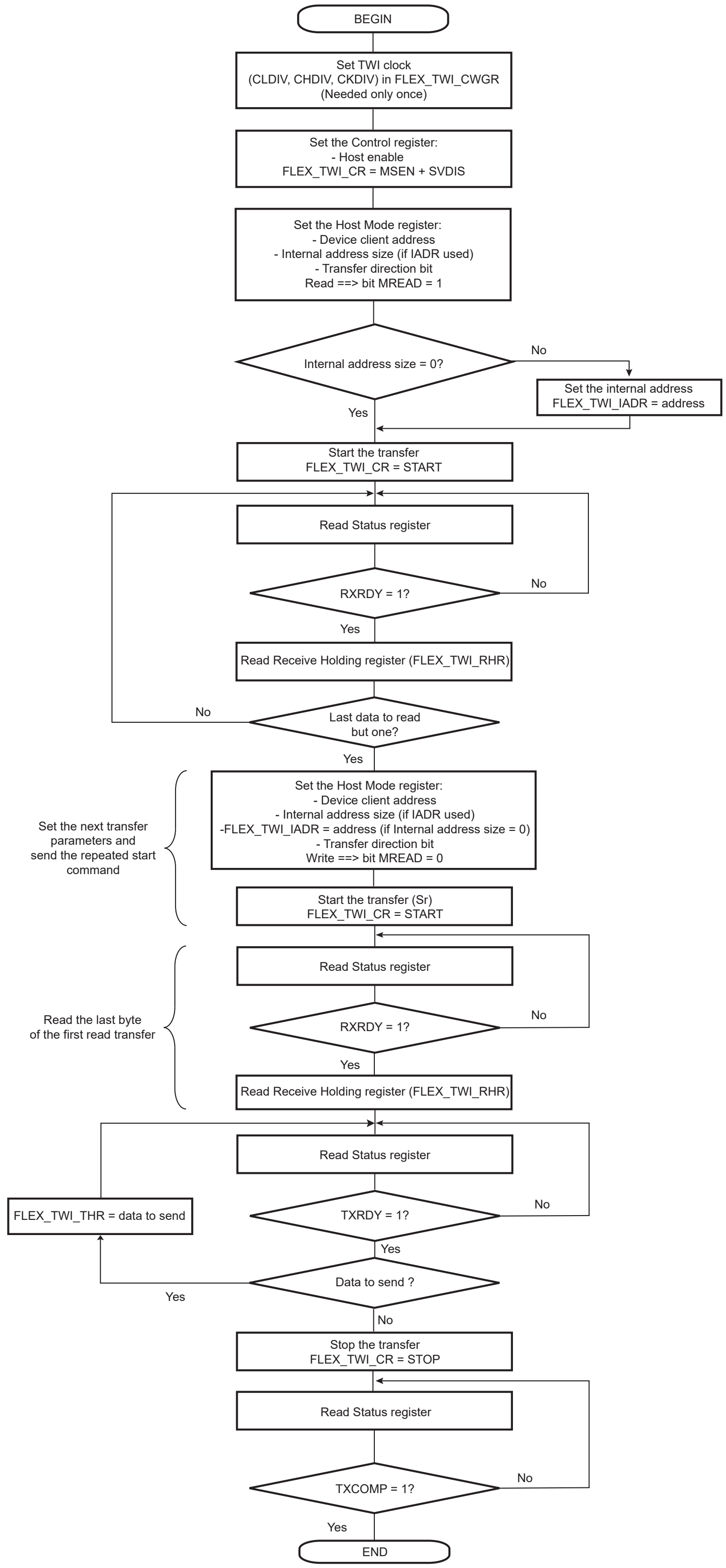
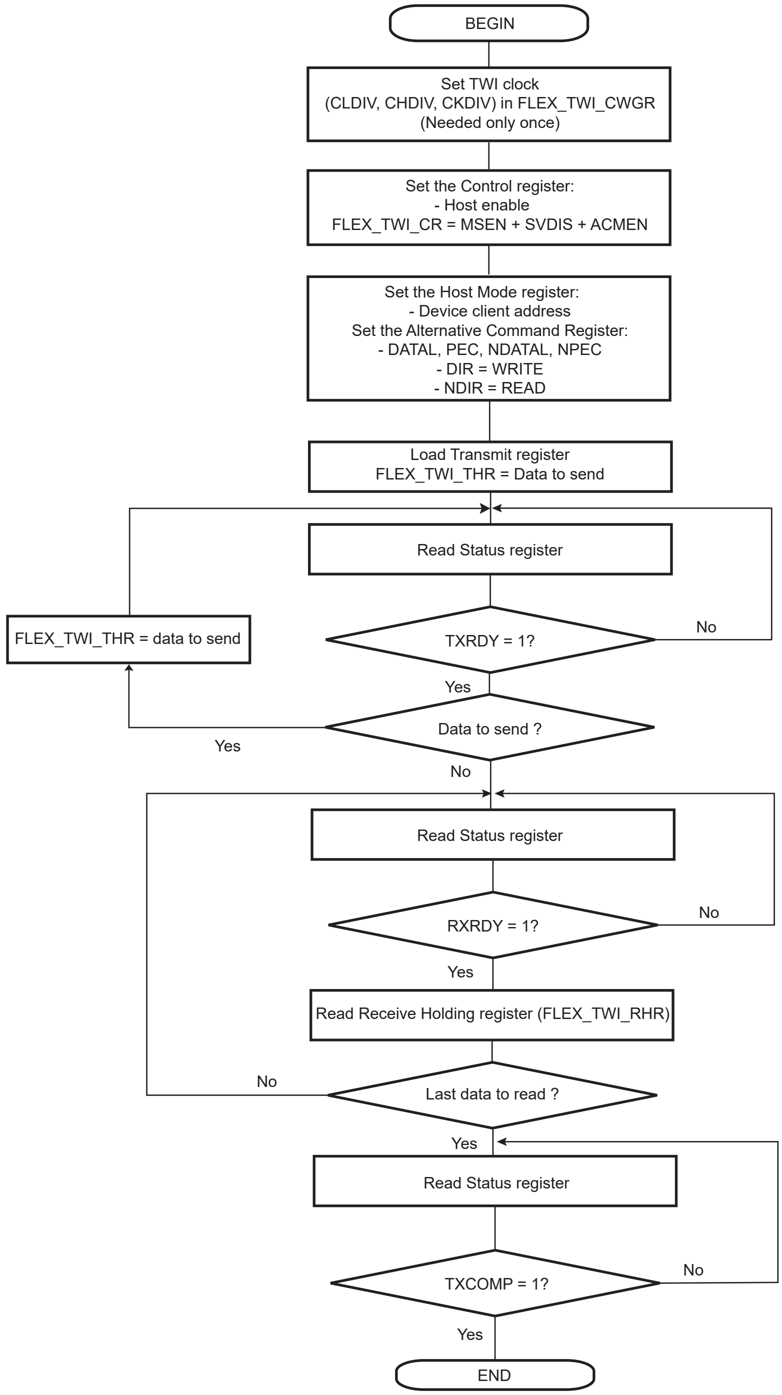


Figure 63-110. TWI Read Operation with Multiple Data Bytes + Write with Alternative Command Mode with PEC



63.9.4 Multi-Host Mode

63.9.4.1 Definition

In Multi-Host mode, more than one host may handle the bus at the same time without data corruption by using arbitration.

Arbitration starts as soon as two or more hosts place information on the bus at the same time, and stops (arbitration is lost) for the host that intends to send a logical one while the other host sends a logical zero.

As soon as arbitration is lost by a host, it stops sending data and listens to the bus in order to detect a STOP. When the STOP is detected, the host that has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in figure "Arbitration Cases" below.

63.9.4.2 Different Multi-Host Modes

Two Multi-Host modes are available:

- TWI as Host Only—TWI is considered as a host only and will never be addressed.
- TWI as Host or Client—TWI may be either a host or a client and may be addressed.

Note: Arbitration in supported in both Multi-Host modes.

63.9.4.2.1 TWI as Host Only

In this mode, the TWI is considered as a host only (MSEN is always at one) and must be driven like a host with the ARBLST (ARBitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If the user starts a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWI automatically waits for a STOP condition on the bus to initiate the transfer (see figure "User Sends Data While the Bus is Busy" below).

Note: The state of the bus (busy or free) is not indicated in the user interface.

63.9.4.2.2 TWI as Host or Client

The automatic reversal from host to client is not supported in case of a lost arbitration.

Then, in the case where TWI may be either a host or a client, the user must manage the pseudo Multi-Host mode described in the steps below:

1. Program the TWI in Client mode (SADR + MSDIS + SVEN) and perform a client access (if TWI is addressed).
2. If the TWI has to be set to Host mode, wait until TXCOMP flag is at 1.
3. Program Host mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as Host mode is enabled, the TWI scans the bus in order to detect if it is busy or free. When the bus is considered as free, the TWI initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is = 1), the user must program the TWI in Client mode in case the host that won the arbitration needs to access the TWI.
7. If the TWI has to be set to Client mode, wait until TXCOMP flag is at 1 and then program Client mode.
   Note: In case the arbitration is lost and the TWI is addressed, the TWI will not acknowledge even if it is programmed in Client mode as soon as ARBLST = 1. Then the host must repeat SADR.

Figure 63-111. User Sends Data While the Bus is Busy

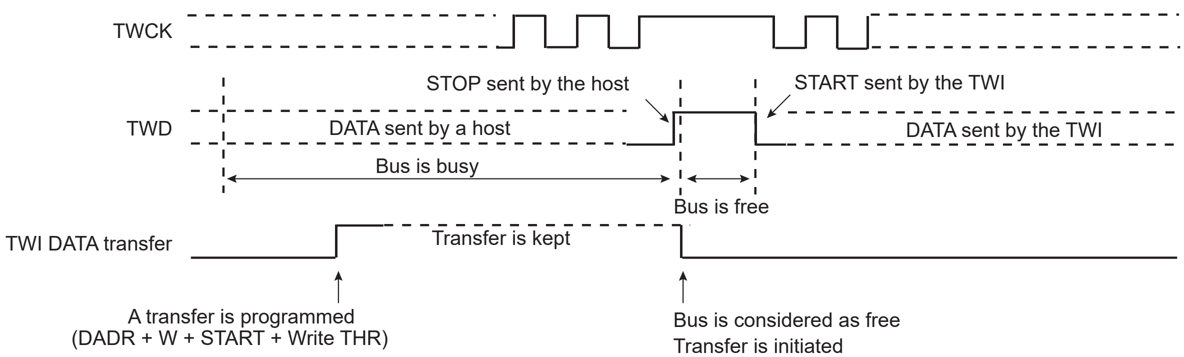
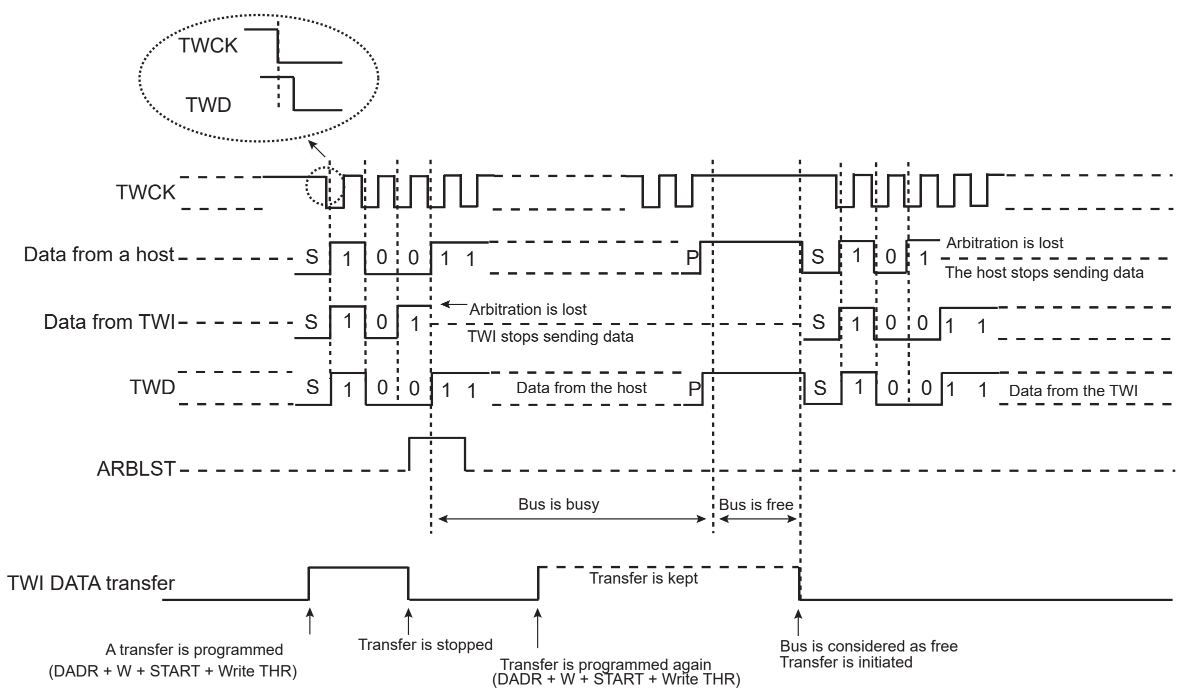
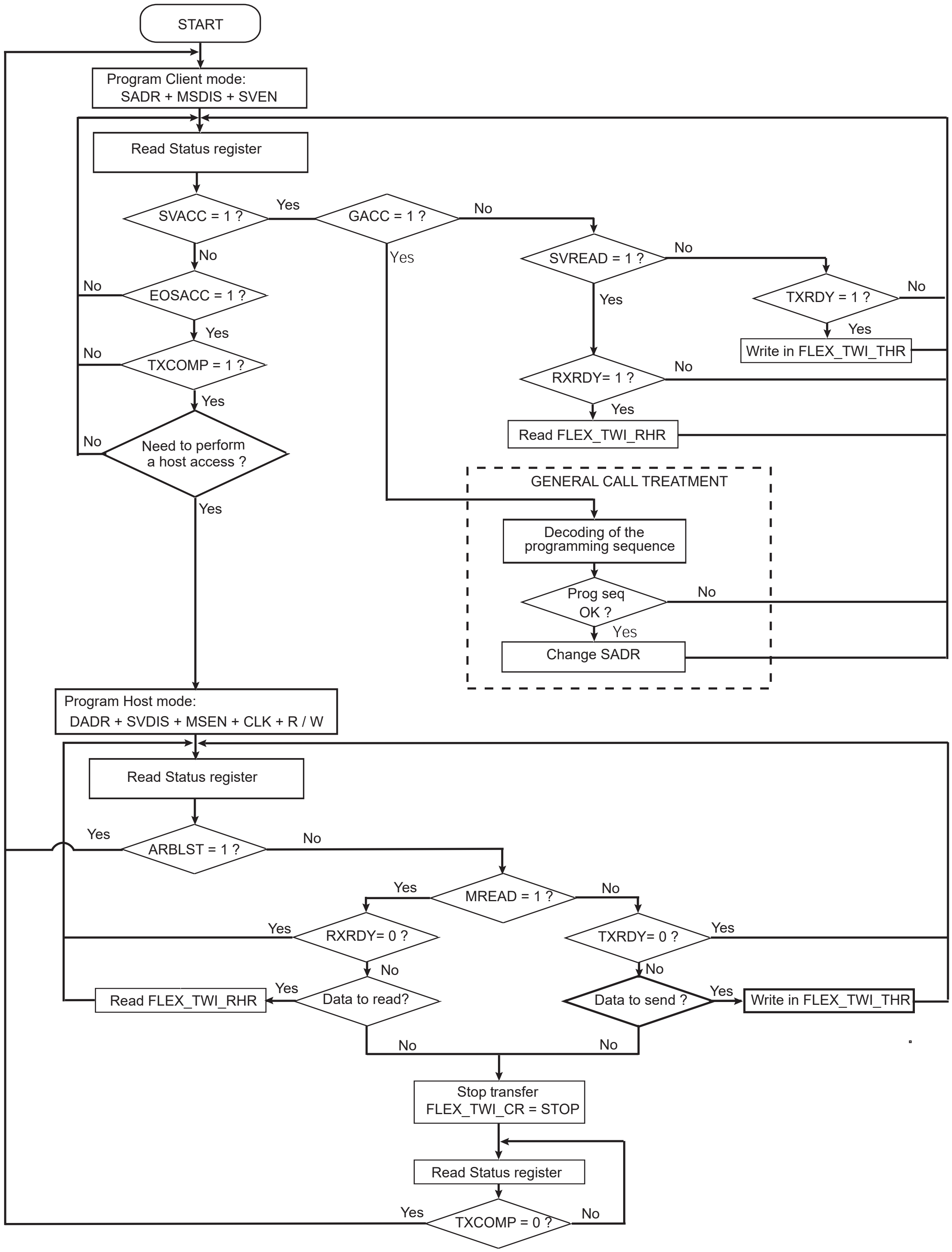


Figure 63-112. Arbitration Cases



The flowchart shown in the following figure gives an example of read and write operations in Multi-Host mode.

Figure 63-113. Multi-Host Mode



63.9.5 Client Mode

63.9.5.1 Definition

Client mode is defined as a mode where the device receives the clock and the address from another device called the host.

In this mode, the device never initiates and never completes the transmission (START, REPEATED_START and STOP conditions are always provided by the host).

63.9.5.2 Programming Client Mode

The following fields must be programmed before entering Client mode:

1. FLEX_TWI_SMR.SADR: The client device address is used in order to be accessed by host devices in Read or Write mode.
2. (Optional) FLEX_TWI_SMR.MASK can be set to mask some SADR address bits and thus allow multiple address matching.
3. FLEX_TWI_CR.MSDIS: Disables Host mode.
4. FLEX_TWI_CR.SVEN: Enables Client mode.

As the device receives the clock, values written in FLEX_TWI_CWGR are not processed.

63.9.5.3 Receiving Data

After a START or repeated START condition is detected, and if the address sent by the host matches the client address programmed in the SADR (Client Address) field, the SVACC (Client Access) flag is set and SVREAD (Client Read) indicates the direction of the transfer.

SVACC remains high until a STOP condition or a repeated START is detected. When such a condition is detected, EOSACC (End Of Client Access) flag is set.

63.9.5.3.1 Read Sequence

In the case of a read sequence (SVREAD is high), the TWI transfers data written in FLEX_TWI_THR (TWI Transmit Holding register) until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the read sequence TXCOMP (Transmission Complete) flag is set and SVACC is reset.

As soon as data is written in FLEX_TWI_THR, the TXRDY (Transmit Holding Register Ready) flag is reset, and it is set when the internal shifter is empty and the sent data acknowledged or not. If the data is not acknowledged, the NACK flag is set.

Note that a STOP or a repeated START always follows a NACK.

See figure "Read Access Ordered by a Host" below.

Note: To clear the TXRDY flag in Client mode, write the FLEX_TWI_CR.SVDIS bit to 1, then write the FLEX_TWI_CR.SVEN bit to 1.

63.9.5.3.2 Write Sequence

In the case of a write sequence (SVREAD is low), the RXRDY (Receive Holding Register Ready) flag is set as soon as a character has been received in FLEX_TWI_RHR (TWI Receive Holding register). RXRDY is reset when reading FLEX_TWI_RHR.

TWI continues receiving data until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the write sequence TXCOMP flag is set and SVACC reset.

See figure "Write Access Ordered by a Host" below.

63.9.5.3.3 Clock Stretching Sequence

If FLEX_TWI_THR or FLEX_TWI_RHR is not written/read in time, the TWI performs a clock stretching.

Clock stretching information is given by the SCLWS (Clock Wait State) bit.

See figures "Clock Stretching in Read Mode" and "Clock Stretching in Write Mode" below.

Note: Clock stretching can be disabled by configuring the FLEX_TWI_SMR.SCLWSDIS bit. In that case, UNRE and OVRE flags will indicate underrun (when FLEX_TWI_THR is not filled on time) or overrun (when FLEX_TWI_RHR is not read on time).

63.9.5.3.4 General Call

In the case where a GENERAL CALL is performed, the GACC (General Call Access) flag is set.

After GACC is set, it is up to the user to interpret the meaning of the GENERAL CALL and to decode the new address programming sequence.

See figure "Host Performs a General Call" below.

63.9.5.4 Data Transfer

63.9.5.4.1 Read Operation

Read mode is defined as a data requirement from the host.

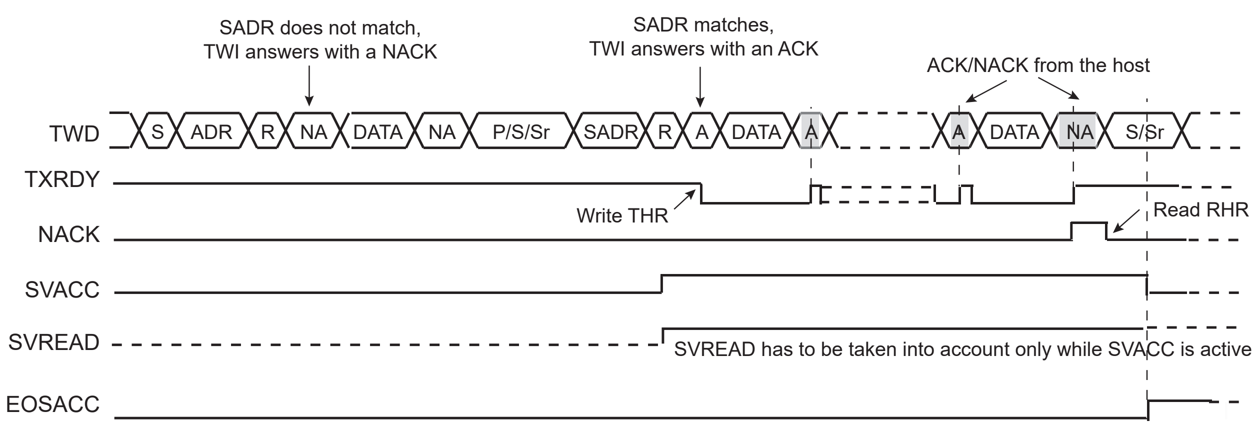
After a START or a REPEATED START condition is detected, the decoding of the address starts. If the client address (SADR) is decoded, SVACC is set and SVREAD indicates the direction of the transfer.

Until a STOP or REPEATED START condition is detected, TWI continues sending data loaded in FLEX_TWI_THR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the read operation.

Figure 63-114. Read Access Ordered by a Host



Notes:
1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. TXRDY is reset when data has been transmitted from FLEX_TWI_THR to the internal shifter and set when this data has been acknowledged or non acknowledged.

63.9.5.4.2 Write Operation

The Write mode is defined as a data transmission from the host.

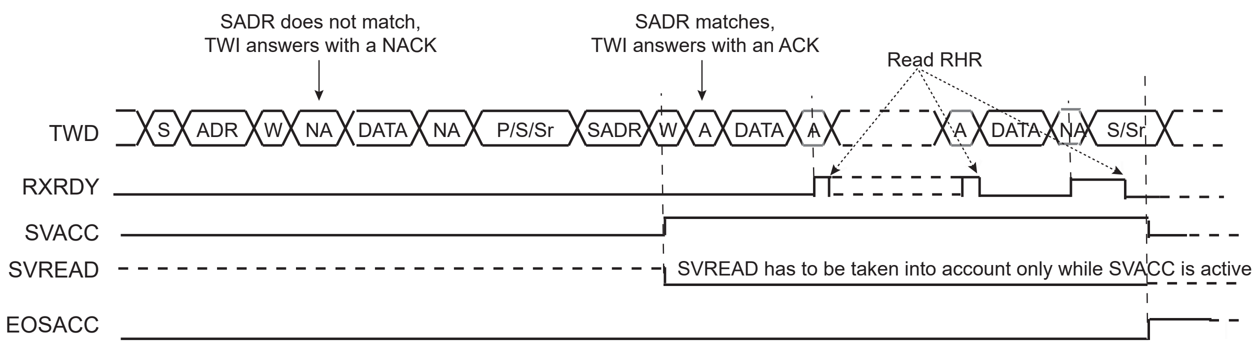
After a START or a REPEATED START, the decoding of the address starts. If the client address is decoded, SVACC is set and SVREAD indicates the direction of the transfer (SVREAD is low in this case).

Until a STOP or REPEATED START condition is detected, TWI stores the received data in FLEX_TWI_RHR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the write operation.

Figure 63-115. Write Access Ordered by a Host



Notes:
1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. RXRDY is set when data has been transmitted from the internal shifter to FLEX_TWI_RHR, and reset when this data is read.

63.9.5.4.3 General Call

The general call is performed in order to change the address of the client.

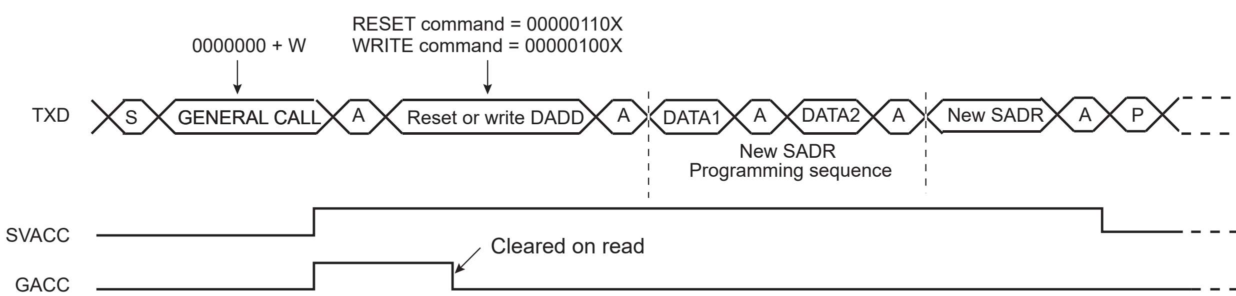
If a GENERAL CALL is detected, GACC is set.

After the detection of general call, it is up to the user to decode the commands which follow.

In case of a WRITE command, the user has to decode the programming sequence and program a new SADR if the programming sequence matches.

The following figure describes the general call access.

Figure 63-116. Host Performs a General Call



Note: This method enables to create a user-specific programming sequence by choosing the number of programming bytes. The programming sequence has to be provided to the host.

63.9.5.4.4 Clock Stretching

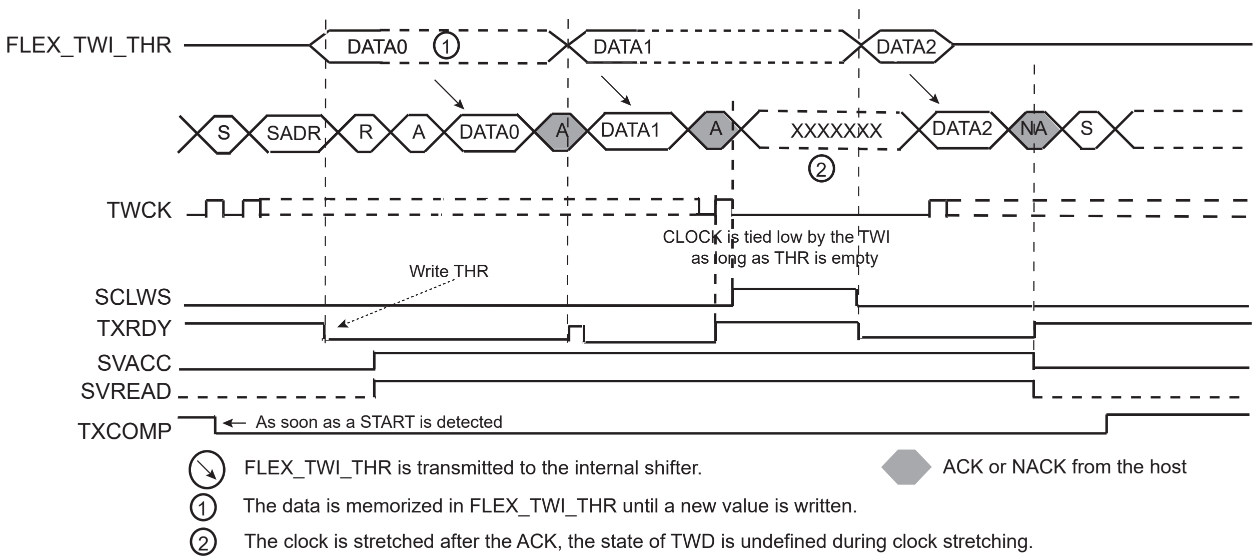
In both Read and Write modes, it may happen that the FLEX_TWI_THR/FLEX_TWI_RHR buffer is not filled/emptied before the transmission/reception of a new character. In this case, to avoid sending/receiving undesired data, a clock stretching mechanism is implemented.

Note: Clock stretching can be disabled by setting the FLEX_TWI_SMR.SCLWSDIS bit. In that case, the UNRE and OVRE flags indicate an underrun (when FLEX_TWI_THR is not filled on time) or an overrun (when FLEX_TWI_RHR is not read on time).

— Clock Stretching in Read Mode
The clock is tied low if the internal shifter is empty and if a STOP or REPEATED START condition was not detected. It is tied low until the internal shifter is loaded.

The following figure describes clock stretching in Read mode.

Figure 63-117. Clock Stretching in Read Mode

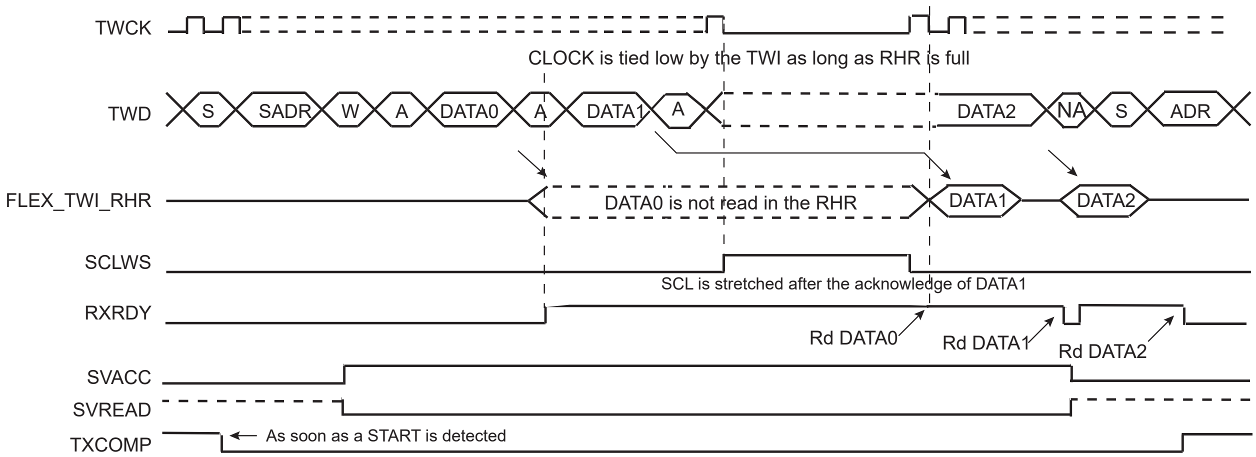


Notes:
1. TXRDY is reset when data has been written in FLEX_TWI_THR to the internal shifter, and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically set when the clock stretching mechanism is started.

— Clock Stretching in Write Mode
The clock is tied low if the internal shifter and FLEX_TWI_RHR are full. If a STOP or REPEATED_START condition was not detected, it is tied low until FLEX_TWI_RHR is read.

The following figure describes the clock stretching in Write mode.

Figure 63-118. Clock Stretching in Write Mode



Notes:

1. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
2. SCLWS is automatically set when the clock stretching mechanism is started and automatically reset when the mechanism is finished.

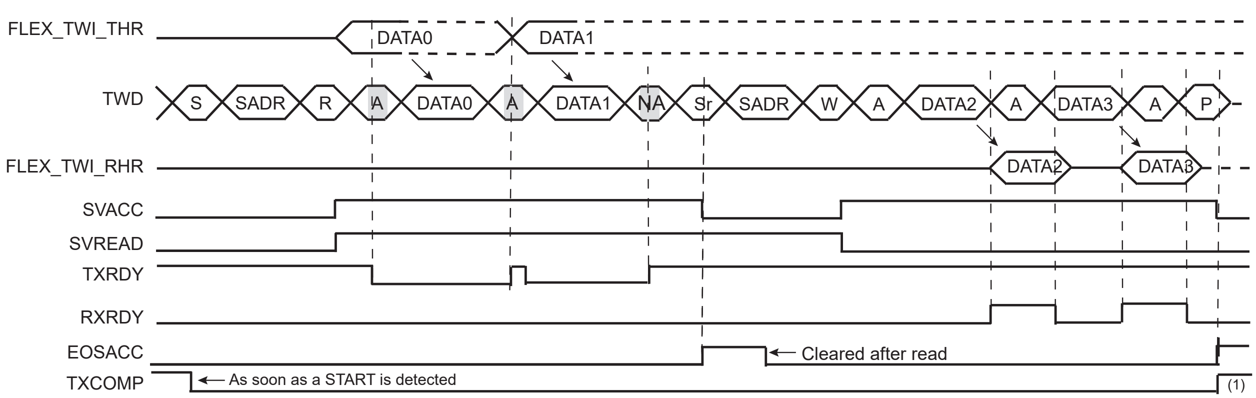
63.9.5.4.5 Reversal after a Repeated Start

— Reversal of Read to Write

The host initiates the communication by a read command and finishes it by a write command.

The following figure describes the repeated start and the reversal from Read mode to Write mode.

Figure 63-119. Repeated Start and Reversal from Read Mode to Write Mode



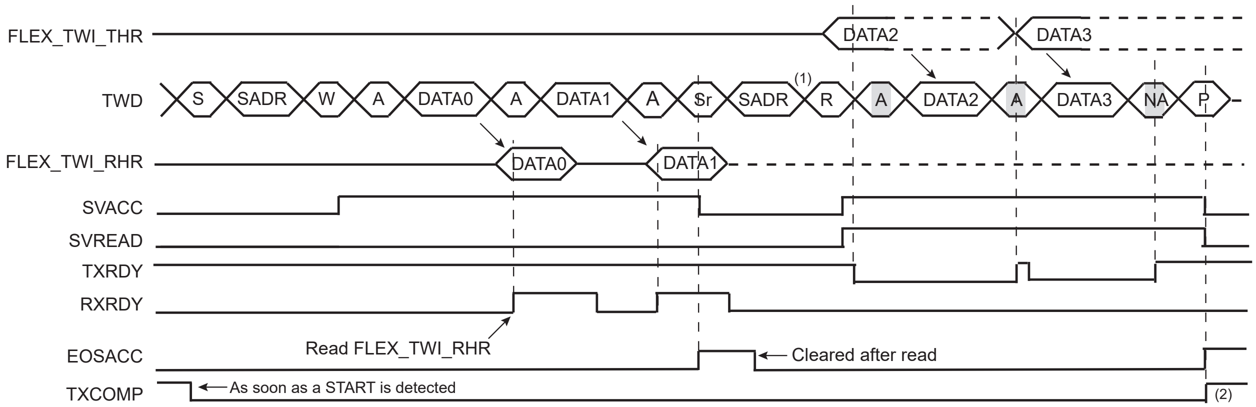
Note:
1. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

— Reversal of Write to Read

The host initiates the communication by a write command and finishes it by a read command.

The following figure describes the repeated start and the reversal from Write mode to Read mode.

Figure 63-120. Repeated Start and Reversal from Write Mode to Read Mode



Notes:
1. In this case, if FLEX_TWI_THR has not been written at the end of the read command, the clock is automatically stretched before the ACK.
2. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

63.9.5.4.6 SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.
2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses have been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

— Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one will send/check the FLEX_TWI_ACR.PEC field in the current transfer. The PEC generator is always updated on every bit transmitted or received, so that PEC handling on following linked transfers will be correct.

In Client Receiver mode, the host calculates a PEC value and transmits it to the client after all data bytes have been transmitted. Upon reception of this PEC byte, the client will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the client will return an ACK to the host. If the PEC values differ, data was corrupted, and the client will return a NACK value. The FLEX_TWI_SR.PECERR bit is set automatically if a PEC error occurred.

In Client Transmitter mode, the client calculates a PEC value and transmits it to the host after all data bytes have been transmitted. Upon reception of this PEC byte, the host will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the host must take appropriate action.

See Client Read/Write Flowcharts for detailed flowcharts.

— Timeouts

The TWI SMBus Timing register (FLEX_TWI_SMBTR) configures the SMBus timeout values. If a timeout occurs, the client leaves the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

63.9.5.5 High-Speed Client Mode

High-speed mode is enabled when the FLEX_TWI_CR.HSEN bit is written to one. Furthermore, the analog pad filter must be enabled, the FLEX_TWI_FILTR.PADFEN bit must be written to one and the FLEX_TWI_FILTR.FILT bit must be cleared. TWI High-speed mode operation is similar to TWI operation with the following exceptions:

1. A host code is received first at normal speed before entering High-speed mode period.

2. When TWI High-speed mode is active, clock stretching is only allowed after acknowledge (ACK), not-acknowledge (NACK), START (S) or repeated START (Sr) (asa consequence, OVF may happen).

TWI High-speed mode allows transfers of up to 3.4 Mbit/s.

The TWI client in High-speed mode requires that the peripheral clock runs at a minimum of 14 MHz if client clock stretching is enabled (SCLWSDIS bit at ‘0’). If client clock stretching is disabled (SCLWSDIS bit at ‘1’), the peripheral clock must run at a minimum of 11 MHz (assuming the system has no latency).

Notes:
1. When client clock stretching is disabled, FLEX_TWI_RHR must always be read before receiving the next data (write access frame generated by the host). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.RXRDY flag, or the DMA. If the receive is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid receive overrun.
2. When client clock stretching is disabled, FLEX_TWI_THR must be filled with the first data to send before the beginning of the frame (read access frame generated by the host). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.TXRDY flag, or the DMA. If the transmit is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid transmit underrun.

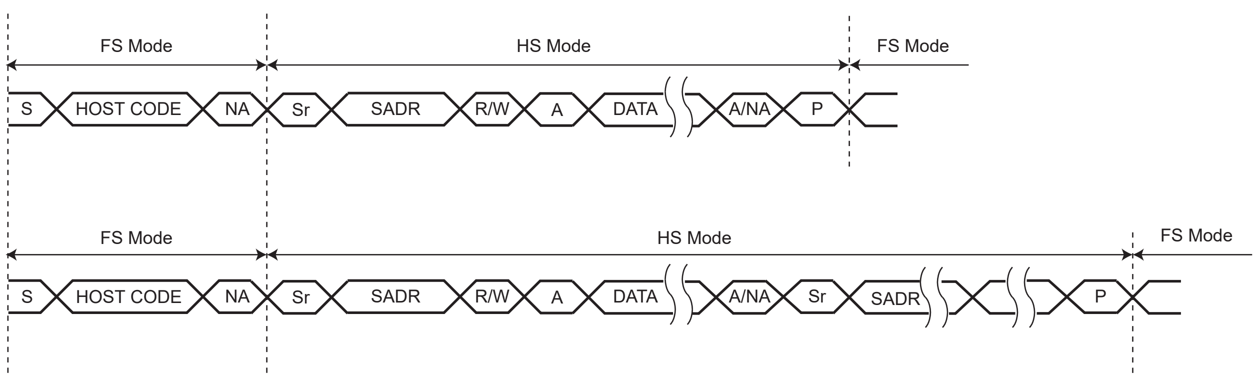
63.9.5.5.1 Read/Write Operation

A TWI high-speed frame always begins with the following sequence:

1. START condition (S)
2. Host Code (0000 1XXX)
3. Not-acknowledge (NACK)

When the TWI is programmed in Client mode and TWI High-speed mode is activated, host code matching is activated and internal timings are set to match the TWI High-speed mode requirements.

Figure 63-121. High-Speed Mode Read/Write



63.9.5.5.2 Usage

TWI High-speed mode usage is the same as the standard TWI (see Read/Write Flowcharts).

63.9.5.6 Alternative Command

In Client mode, the Alternative Command mode is used when the SMBus mode is enabled to send or check the PEC byte.

The Alternative Command mode is enabled by setting the ACMEN bit of the TWIHS Control register, and the transfer is configured in TWIHS_ACR.

For a combined transfer with PEC, only the NPEC bit in TWIHS_ACR must be set as the PEC byte is sent once at the end of the frame.

See Client Read/Write Flowcharts for detailed flowcharts.

63.9.5.7 TWI Asynchronous and Partial Wakeup

The TWI module includes an asynchronous start condition detector, capable of waking the device up from a Sleep mode upon an address match (and optionally an additional data match), including Sleep modes where the TWI peripheral clock is stopped. It can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

FLEX_TWI_RHR must be read before enabling the asynchronous and partial wakeup.

After detecting the START condition on the bus, the TWI will stretch TWCK until the TWI peripheral clock has started. The time required for starting the TWI peripheral depends on which Sleep mode the device is in. After the TWI peripheral clock has started, the TWI releases its TWCK stretching and receives one byte of data (client address) on the bus. At this time, only a limited part of the device, including the TWI module, receives a clock, thus saving power. If the address phase causes a TWIS address match (and optionally if the first data byte causes data match as well), the entire device is wakened and normal TWI address matching actions are performed. Normal TWI transfer then follows. If the TWI module is not addressed (or if the optional data match fails), the TWI peripheral clock is automatically stopped and the device returns to its original Sleep mode.

The TWI module has the capability to match on more than one address. The FLEX_TWI_SMR.SADR1EN/ SADR2EN/SADR3EN bits enable address matching on additional addresses which can be configured through the FLEX_TWI_SWMR.SADR1/SADR2/SADR3 fields. The matching process can be extended to the first received data byte if the FLEX_TWI_SMR.DATAMEN bit is set. In that case, a complete matching includes address matching and first received data matching. The FLEX_TWI_SWMR.DATAM field can be used to configure the data to match on the first received byte.

When the system is in Active mode and the TWI enters asynchronous partial Wakeup mode, the flag SVACC must be programmed as the unique source of the TWI interrupt and the data match comparison must be disabled.

When the system exits Wait mode as the result of a matching condition, the SVACC flag is used to determine if the TWI is the source of the exit from Wait mode.

Figure 63-122. Address Match and Data Matching Disabled

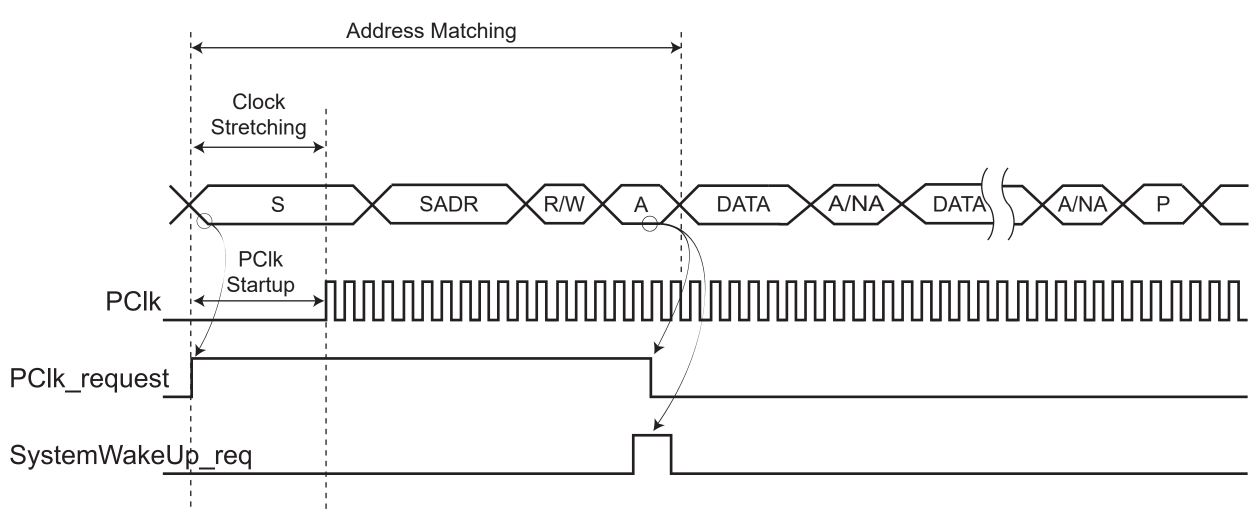


Figure 63-123. Address Does Not Match and Data Matching Disabled

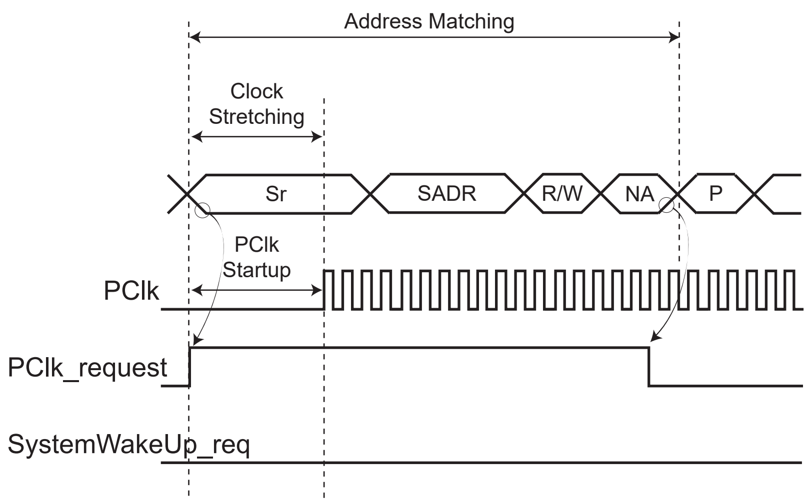


Figure 63-124. Address and Data Match (Data Matching Enabled)

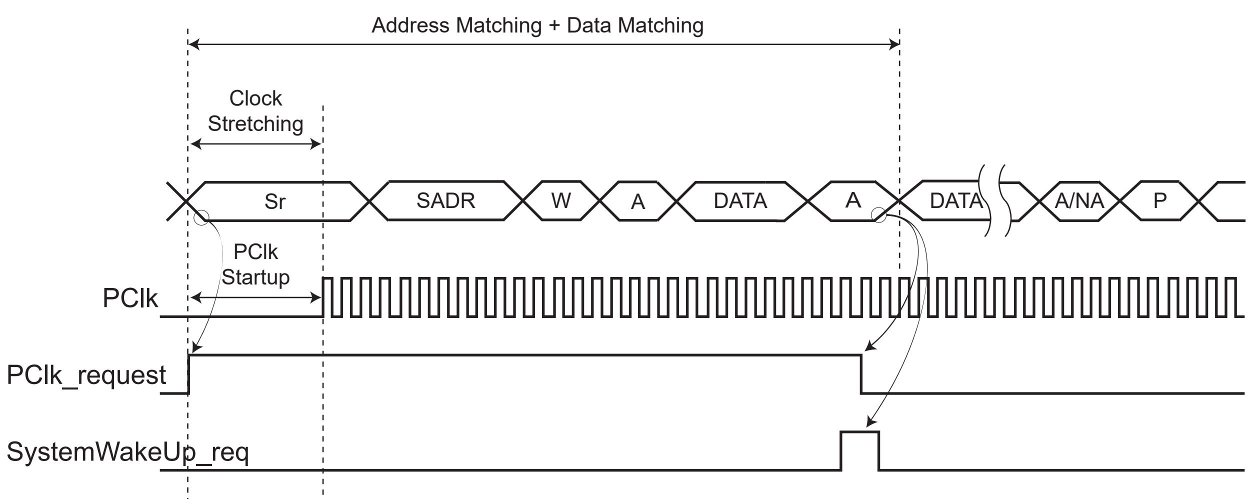
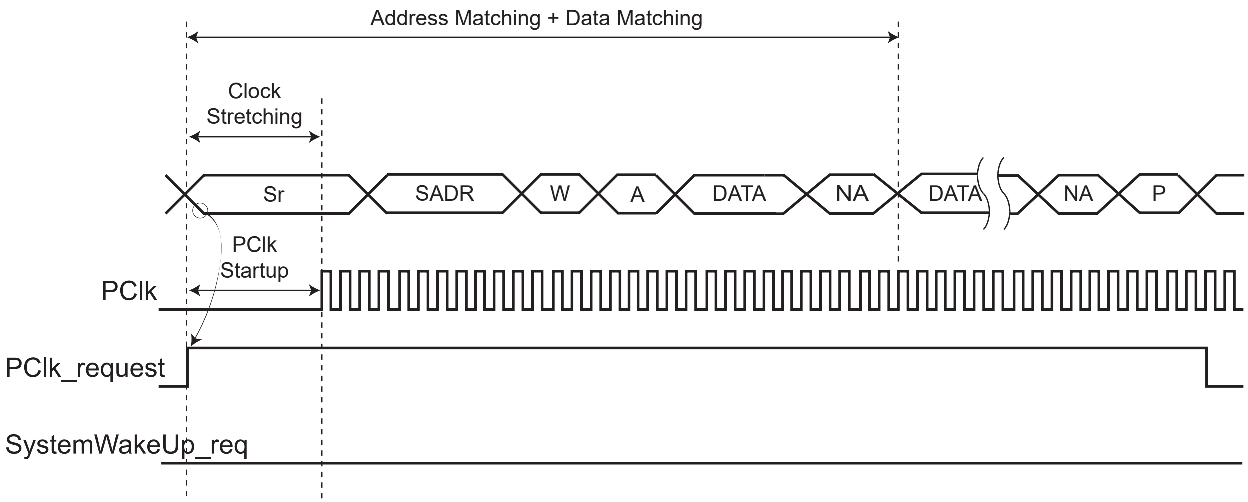


Figure 63-125. Address Matches and Data Do Not Match (Data Matching Enabled)



63.9.5.8 Client Read/Write Flowcharts

The flowchart shown in the following figure gives an example of read and write operations in Client mode. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable register (FLEX_TWI_IER) be configured first.

Figure 63-126. Read/Write in Client Mode

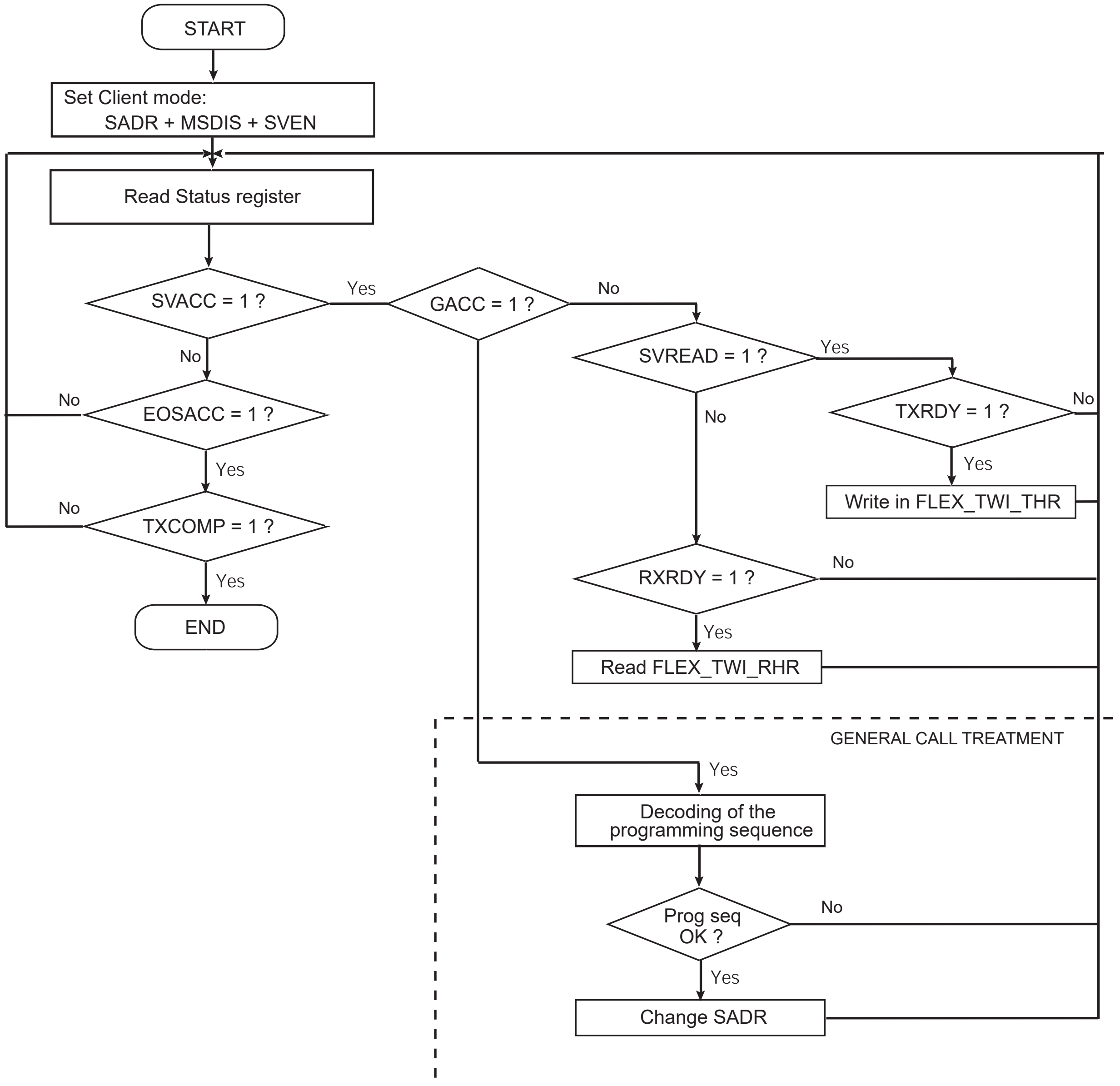


Figure 63-127. Read/Write in Client Mode with SMBus PEC

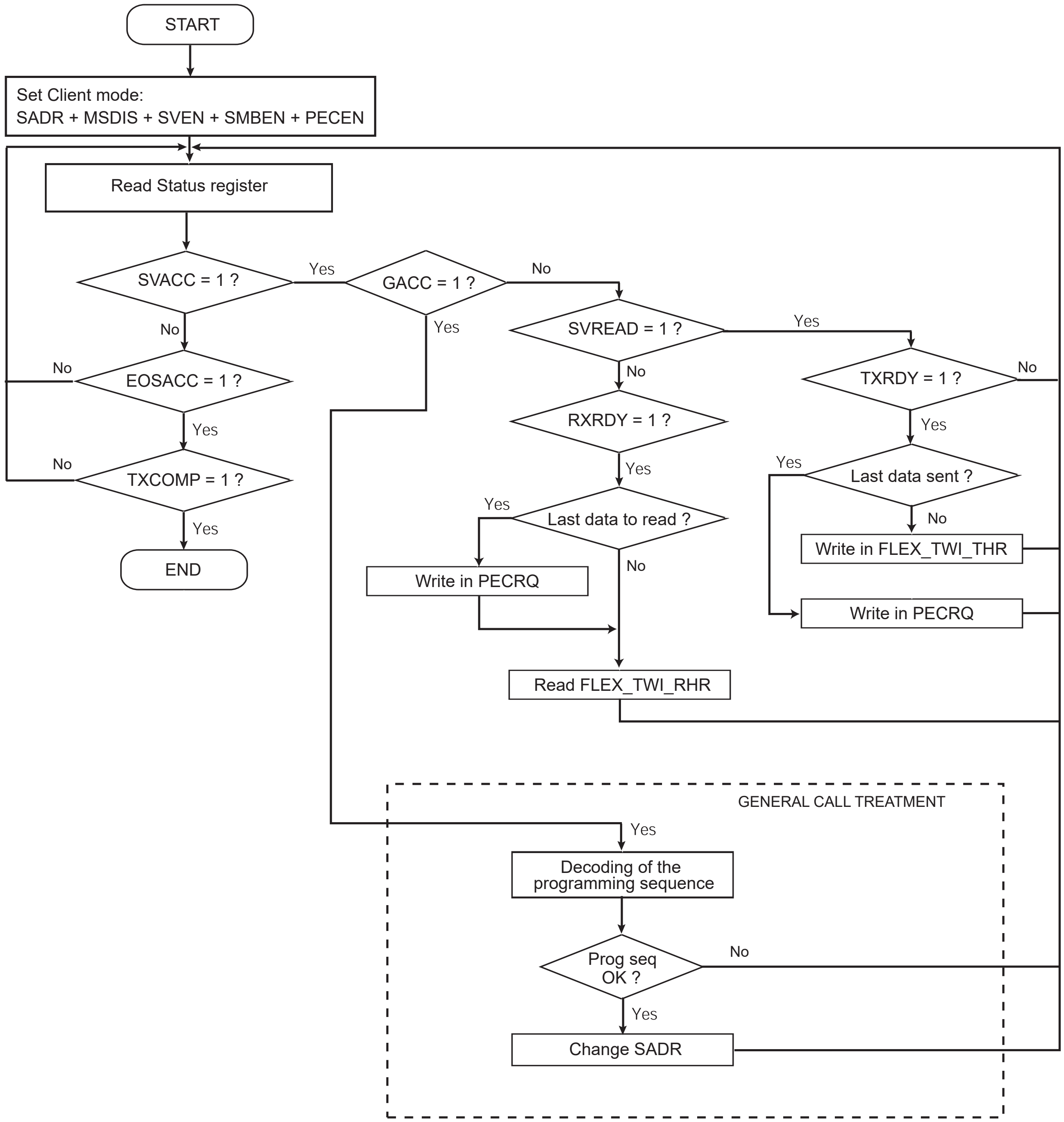
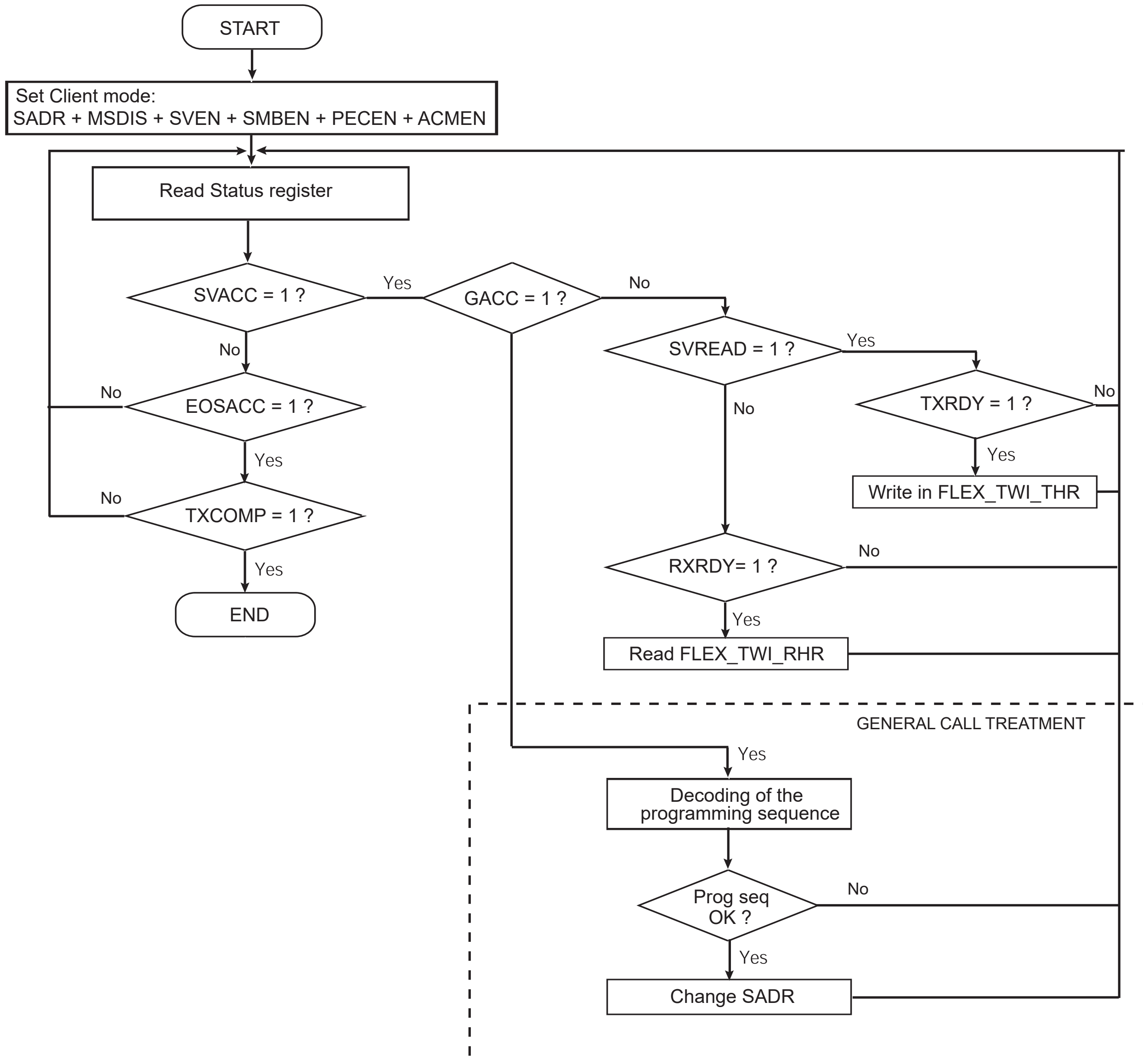


Figure 63-128. Read/Write in Client Mode with SMBus PEC and Alternative Command Mode



63.9.6 TWI FIFOs

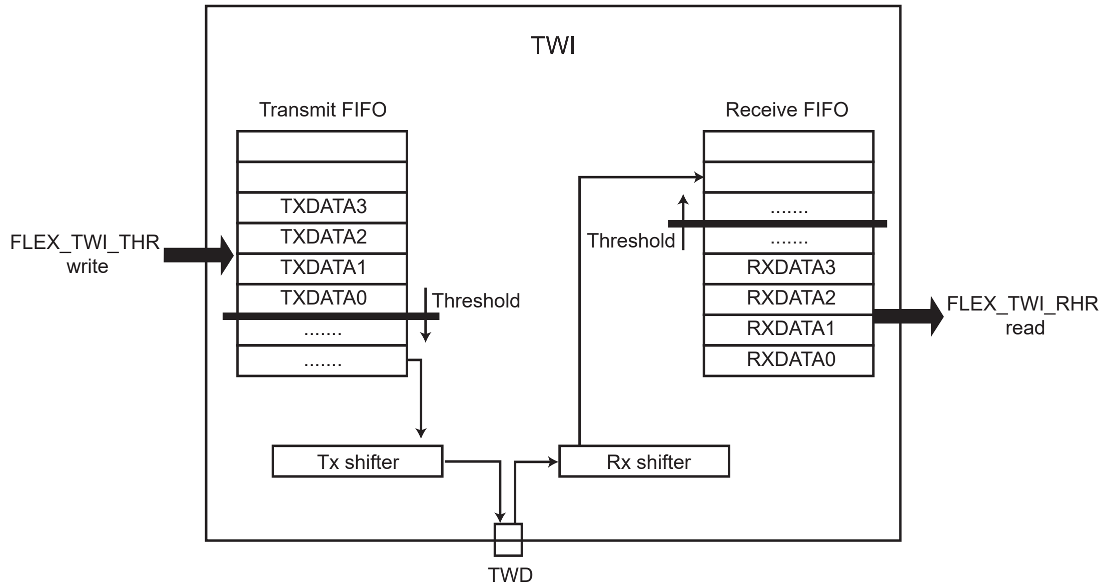
63.9.6.1 Overview

The TWI includes two FIFOs which can be enabled/disabled using FLEX_TWI_CR.FIFOEN/FIFODIS. Both Host and Client modes must be disabled before enabling or disabling the FIFOs (FLEX_TWI_CR.MSDIS/SVDIS).

Writing FLEX_TWI_CR.FIFOEN to ‘1’ enables a 32-byte Transmit FIFO and a 32-byte Receive FIFO.

It is possible to write or to read single or multiple bytes in the same access to FLEX_TWI_THR/RHR, depending on FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

Figure 63-129. TWI FIFOs Block Diagram



63.9.6.2 Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_TWI_THR loads the Transmit FIFO.

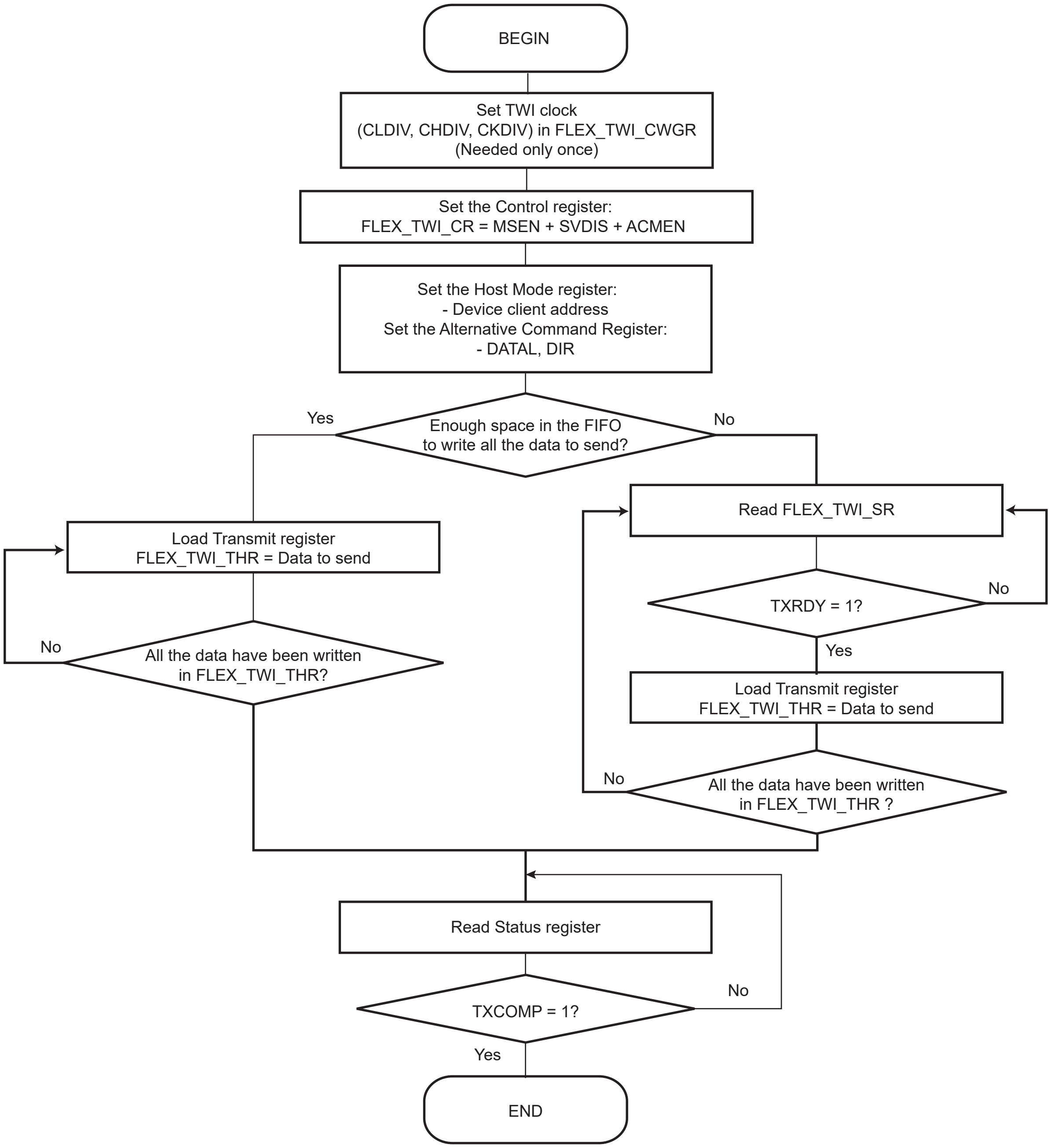
The Transmit FIFO level is provided in FLEX_TWI_FLR.TXFL. If the FIFO can accept the number of bytes to be transmitted, there is no need to monitor FLEX_TWI_SR.TXRDY and the bytes can be successively written in FLEX_TWI_THR.

If the FIFO cannot accept the bytes due to insufficient space, wait for the TXRDY flag to be set before writing the bytes in FLEX_TWI_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

See figures Sending Data with FIFO Enabled in Host Mode and Sending/Receiving Data with FIFO Enabled in Client Mode.

Figure 63-130. Sending Data with FIFO Enabled in Host Mode



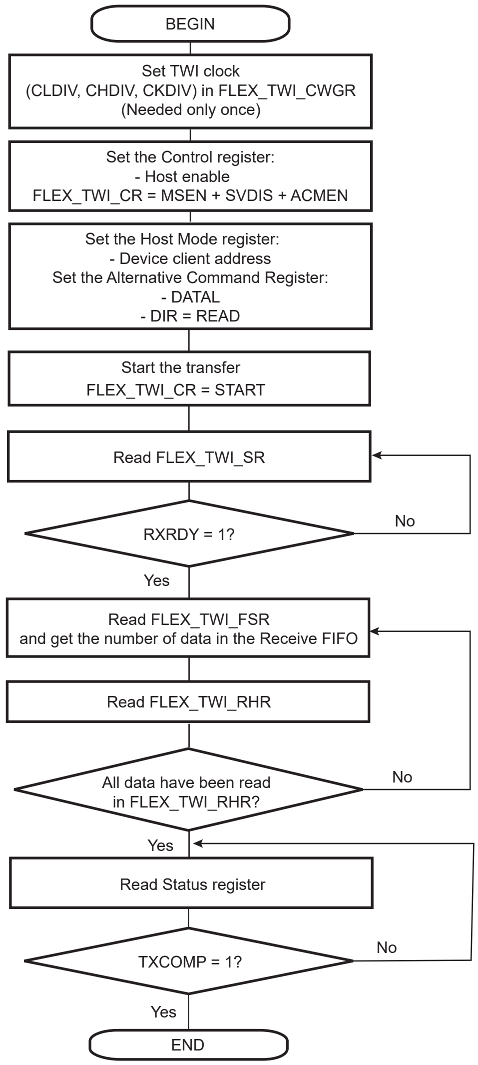
63.9.6.3 Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_TWI_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to ‘1’), the exact number of bytes can be checked with FLEX_TWI_FLR.RXFL. All the bytes can be read successively in FLEX_TWI_RHR without checking the FLEX_TWI_SR.RXRDY flag between each access.

See figures Receiving Data with FIFO Enabled in Host Mode and Sending/Receiving Data with FIFO Enabled in Client Mode.

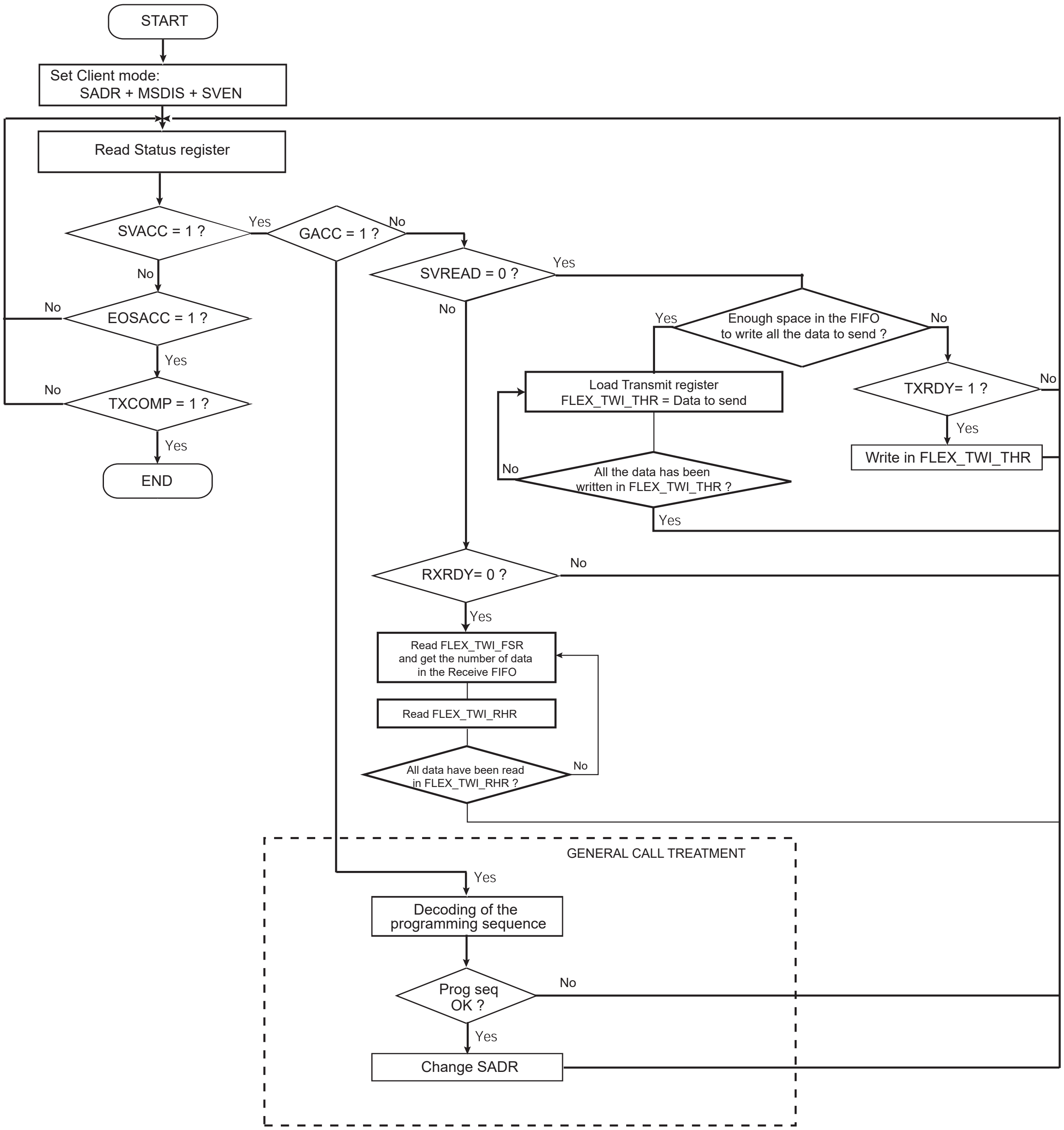
Figure 63-131. Receiving Data with FIFO Enabled in Host Mode



63.9.6.4 Sending/Receiving with FIFO Enabled in Client Mode

See Sending Data with FIFO Enabled and Receiving Data with FIFO Enabled for details.

Figure 63-132. Sending/Receiving Data with FIFO Enabled in Client Mode



63.9.6.5 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_TWI_CR.TXFCLR/RXFCLR.

63.9.6.6 TXRDY and RXRDY Behavior

FLEX_TWI_SR.TXRDY/RXRDY flags display a specific behavior when FIFOs are enabled.

TXRDY indicates if a byte can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new byte. See figure TXRDY Behavior when TXRDYM = 0 in Host Mode.

RXRDY indicates if an unread byte is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread byte is in the Receive FIFO. See figure RXRDY Behavior when RXRDYM = 0 in Host and Client Modes.

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the TWI FIFO Mode register (FLEX_TWI_FMR) to reduce the number of accesses to FLEX_TWI_THR/RHR.

As an example, in Host mode, the Transmit FIFO can be loaded with multiple bytes in the same access by configuring TXRDYM>0.

See FLEX_TWI_FMR for the FIFO configuration.

Figure 63-133. TXRDY Behavior when TXRDYM = 0 in Host Mode

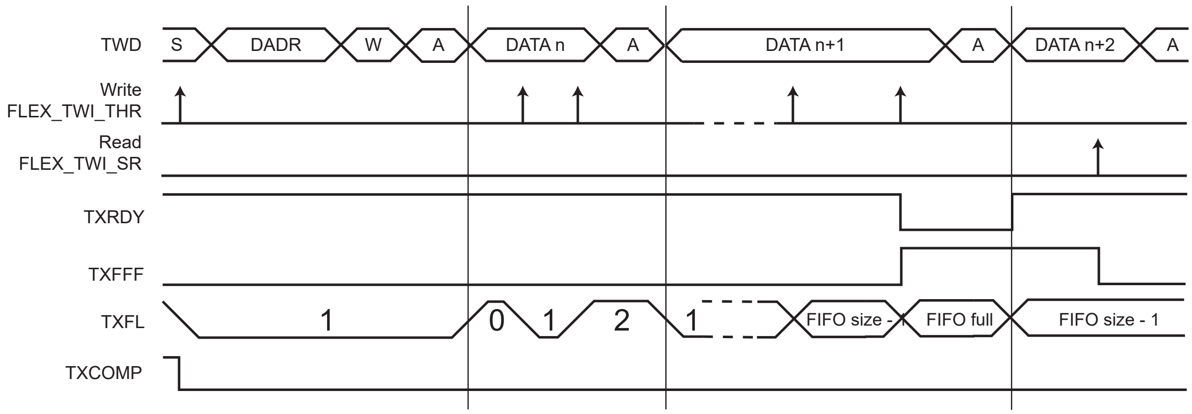


Figure 63-134. RXRDY Behavior when RXRDYM = 0 in Host and Client Modes

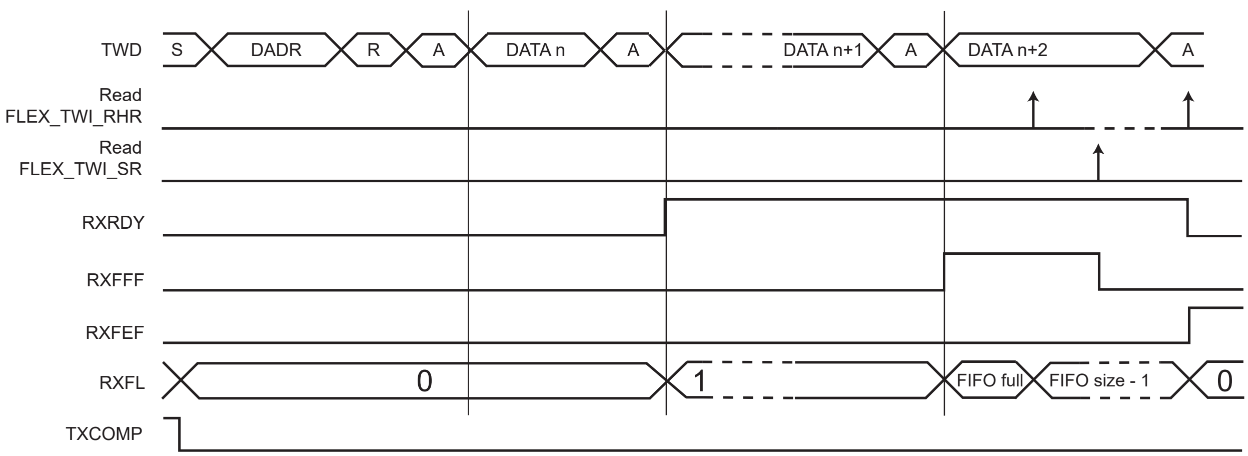
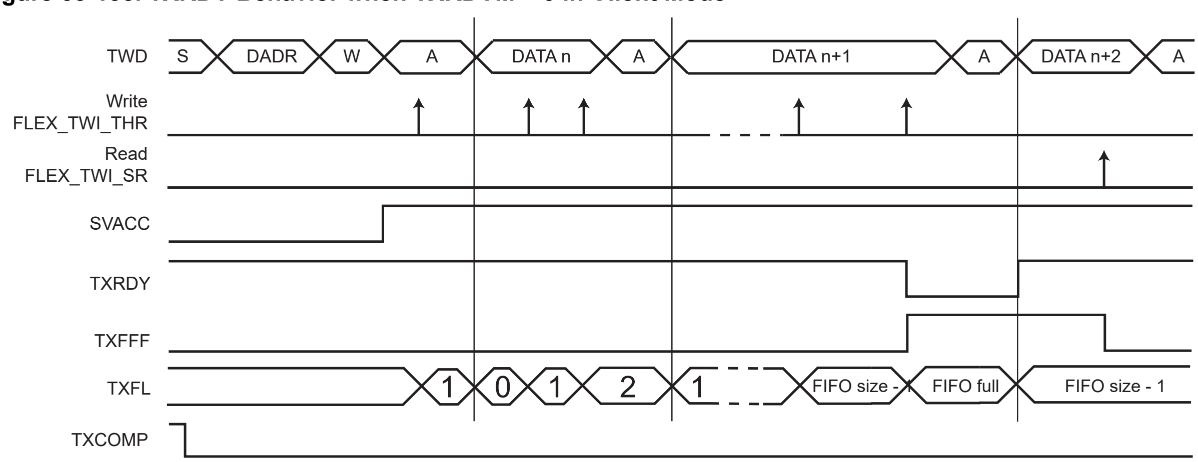


Figure 63-135. TXRDY Behavior when TXRDYM = 0 in Client Mode



63.9.6.7 TWI Single Data Access

When FIFO is enabled and a byte access is performed in FLEX_TWI_THR, one byte is written in the FIFO. The same behavior applies for FLEX_TWI_RHR.

See TWI Transmit Holding Register and TWI Receive Holding Register.

However, it is possible to write/read multiple data each time FLEX_THR_THR/FLEX_US_RHR is accessed. See TWI Multiple Data Access.

63.9.6.8 TWI Multiple Data Access

It is possible to reduce the number of accesses to/from FLEX_TWI_THR/FLEX_US_RHR required to transfer an amount of data, by concatenating multiple bytes.

Up to four data can be written/read in one FLEX_TWI_THR/FLEX_TWI_RHR access when the FIFO is enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode is disabled (FLEX_TWI_SMR.SNIFF=0).

When the FIFO is enabled, the number of bytes to write/read is defined by the type of access in the holding register. If the access is a byte, only one byte is written/read (single data access), if the access is a halfword or a word a multiple data access is performed. If the access is a halfword, then two bytes are written/read and if the access is a word, four bytes are written/read.

Written/Read data are always right-aligned, as described in sections TWI Receive Holding Register (FIFO Enabled) and TWI Transmit Holding Register (FIFO Enabled).

As an example, if the Transmit FIFO is empty and there are six bytes to send, either of the following write accesses may be performed:

- Six FLEX_TWI_THR-byte write accesses
- Three FLEX_TWI_THR-halfword write accesses
- One FLEX_TWI_THR-word write access and one FLEX_TWI_THR halfword write access

With a Receive FIFO containing six bytes, any of the following read accesses may be performed:

- Six FLEX_TWI_RHR-byte read accesses
- Three FLEX_TWI_RHR-halfword read accesses
- One FLEX_TWI_RHR-word read access and one FLEX_TWI_RHR-halfword read access

63.9.6.8.1 TXRDY and RXRDY Configuration

It is possible to write one or more bytes in the same FLEX_TWI_THR/FLEX_TWI_RHR access. The TXRDY flag indicates if one or more bytes can be written in the FIFO depending on the configuration of FLEX_TWI_FMR.TXRDYM/RXRDYM.

When two bytes are written for each FLEX_TWI_THR access, the TXRDYM field can be configured so that the TXRDY flag is at ‘1’ only when at least two bytes can be written in the Transmit FIFO.

When four bytes are read for each FLEX_TWI_RHR access, the RXRDYM field can be configured so that the RXRDY flag is at ‘1’ only when at least four unread bytes are in the Receive FIFO.

63.9.6.8.2 DMAC

The DMAC transfer type must be configured according to the FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

As example, FLEX_TWI_FMR.TXRDYM/RXRDYM=0 is not compatible with DMAC transfers in word (32-bit).

63.9.6.9 Transmit FIFO Lock

If a frame is terminated early due to a not-acknowledge error (NACK flag), SMBus timeout error (TOUT flag) or host code acknowledge error (MACK flag), a lock is set on the Transmit FIFO preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMAC channels, etc., without any risk.

FLEX_TWI_SR.LOCK is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared by setting FLEX_TWI_CR.TXFLCLR to ‘1’.

63.9.6.10 FIFO Pointer Error

A FIFO overflow is reported in FLEX_TWI_FSR.

If the Transmit FIFO is full and a write access is performed on FLEX_TWI_THR, it generates a Transmit FIFO pointer error and sets FLEX_TWI_FSR.TXFPTEF.

If the number of data written in FLEX_TWI_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_TWI_FSR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_TWI_FSR.

If the number of bytes read in FLEX_TWI_RHR (according to the register access size) is greater than the number of unread bytes in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_TWI_FSR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_TWI_THR/FLEX_TWI_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit or Receive pointer error occurs, a software reset must be performed using FLEX_TWI_CR.SWRST. Note that issuing a software while transmitting may leave a client in an unknown state holding the TWD line. In such case, a Bus Clear Command allows the client to release the TWD line (the first frame sent subsequently may not be received properly by the client).

63.9.6.11 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of bytes in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of bytes currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_TWI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from 'above threshold' to 'equal to or below threshold', the flag FLEX_TWI_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_TWI_FMR.RXFTHRES. Each time the Receive FIFO level goes from 'below threshold' to 'equal to or above threshold', the flag FLEX_TWI_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_TWI_FIER and FLEX_TWI_FIDR.

63.9.6.12 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_TWI_FIER and FLEX_TWI_FIDR.

FIFO flags state can be read in FLEX_TWI_FSR. They are cleared when FLEX_TWI_FSR is read.

63.9.7 TWI Comparison Function on Received Character

The comparison function differs if the asynchronous partial wakeup is enabled or not.

If asynchronous partial wakeup is disabled, the TWI has the capability to extend the address matching on up to three client addresses. The FLEX_TWI_SMR.SADR1EN/SADR2EN/SADR3EN bits enable address matching on additional addresses which can be configured through the FLEX_TWI_SWMR.SADR1/SADR2/SADR3 fields. The DATAMEN bit has no effect.

The SVACC bit is set when there is a comparison match with the received client address.

63.9.8 Sniffer Mode

The Client Sniffer mode of a TWI can be enabled to ease the analysis/debug of a TWI bus activity. The TWI bus to be analyzed can be monitored by one TWI host embedded in the product or by an I2C host outside the product.

In this mode, the TWI reports all (or part of) the TWI bus activity without impacting the TWI bus (no bus drive is performed). Depending on the MASK field value, only some specific transfers can be logged instead of the whole activity.

The peripheral TWIn+1 can be configured to analyze the peripheral TWIn (n being the instance index of the TWI) in a full transparent mode via predefined internal connections between TWI instances.

The predefined internal connections provides the capability to use the TWD and TWCK pins for alternate functions while the TWI peripheral is configured in Sniffer Client mode and the selected TWI bus to analyze is carried on these internal links.

The following fields must be programmed before entering Client mode:

1. FLEX_TWI_SMR.SADR: Use the client device address to indicate which frame(s) to log.
2. FLEX_TWI_SMR.MASK: Indicate which SADR bits should be masked and thus which transfers should be logged (set to 0x7F to log the whole TWI bus activity; all SADR bits are masked in this case). General Call accesses will always match.
3. FLEX_TWI_SMR.BSEL: Select the TWI bus to analyze (see figure Sniffer Mode Application Overview).
4. FLEX_TWI_SMR.SNIFF: Set to '1' to enable Client Sniffer mode.
5. FLEX_TWI_CR.MSDIS: Disable Host mode.
6. FLEX_TWI_CR.SVEN: Enable Client mode.

As the device receives the clock, values written in FLEX_TWI_CWGR are not relevant.

Once configured in Client Sniffer mode, the FLEX_TWI_SR.RXRDY bit indicates when a transfer has been logged in FLEX_TWI_RHR. An interrupt can be generated if configured. The FLEX_TWI_SR.OVRE flag indicates if an overrun error occurred if the application is not fast enough to read FLEX_TWI_RHR.

In Client Sniffer mode, FLEX_TWI_RHR logs data as follows:

- The RXDATA field reports the sniffed 8-bit data field.
- The SSTATE field indicates if a START condition has been detected before the 8-bit data field.
- The PSTATE field indicates if a STOP condition has been detected after the previously sniffed 8-bit data field.
- The ASTATE field indicates which acknowledge condition has been detected after the previously sniffed 8-bit data field.

Figure 63-136. Sniffer Mode Application Overview

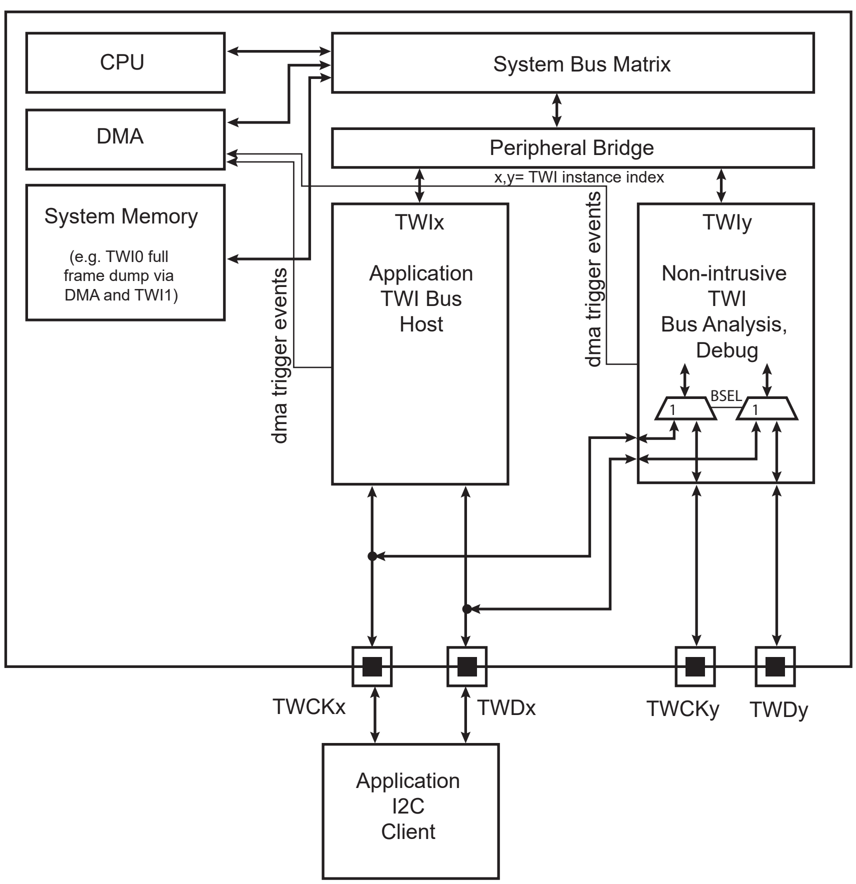
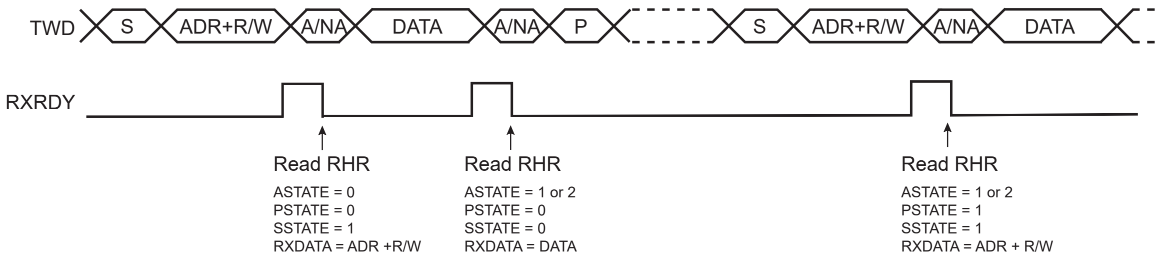


Figure 63-137. Client Sniffer Mode Log



63.9.9 TWI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_TWI to enable access to the write protection registers.

To prevent any single software error from corrupting TWI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TWI Write Protection Mode Register (FLEX_TWI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the TWI Write Protection Status Register (FLEX_TWI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_TWI_WPSR.

The following register(s) can be write-protected when WPEN is set:

- TWI Client Mode Register
- TWI Clock Waveform Generator Register
- TWI SMBus Timing Register
- TWI Matching Register
- TWI FIFO Mode Register

The following register(s) can be write-protected when WPITEN is set:

- TWI Interrupt Enable Register
- TWI Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- TWI Control Register

63.10 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	FLEX_MR	31:24								
		23:16								
		15:8								
		7:0							OPMODE[1:0]	
0x04 ... 0x0F	Reserved									
0x10	FLEX_RHR	31:24								
		23:16								
		15:8	RXDATA[15:8]							
		7:0	RXDATA[7:0]							
0x14 ... 0x1F	Reserved									
0x20	FLEX_THR	31:24								
		23:16								
		15:8	TXDATA[15:8]							
		7:0	TXDATA[7:0]							
0x24 ... 0x01FF	Reserved									
0x0200	FLEX_US_CR	31:24	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
		23:16			LINWKUP	LINABT	RTSDIS	RTSEN		
		15:8	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x0204	FLEX_US_MR	31:24	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
		23:16	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
		15:8	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
		7:0	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
0x0208	FLEX_US_IER	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0208	FLEX_US_IER (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x020C	FLEX_US_IDR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x020C	FLEX_US_IDR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0210	FLEX_US_IMR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0210	FLEX_US_IMR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0214	FLEX_US_CSR	31:24								MANE
		23:16	CTS	CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0214	FLEX_US_CSR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16	LINBLS							
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0218	FLEX_US_RHR	31:24								
		23:16								
		15:8	RXSYNH							RXCHR[8]
		7:0	RXCHR[7:0]							
0x0218	FLEX_US_RHR (FIFO_MULTI_DATA)	31:24	RXCHR3[7:0]							
		23:16	RXCHR2[7:0]							
		15:8	RXCHR1[7:0]							
		7:0	RXCHR0[7:0]							
0x021C	FLEX_US_THR	31:24								
		23:16								
		15:8	TXSYNH							TXCHR[8]
		7:0	TXCHR[7:0]							
0x021C	FLEX_US_THR (FIFO_MULTI_DATA)	31:24	TXCHR3[7:0]							
		23:16	TXCHR2[7:0]							
		15:8	TXCHR1[7:0]							
		7:0	TXCHR0[7:0]							
0x0220	FLEX_US_BRGR	31:24								
		23:16						FP[2:0]		
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x0224	FLEX_US_RTOR	31:24								
		23:16								TO[16]
		15:8	TO[15:8]							
		7:0	TO[7:0]							
0x0228	FLEX_US_TTGR	31:24								
		23:16								
		15:8								
		7:0	TG[7:0]							
0x022C ... 0x023F	Reserved									
0x0240	FLEX_US_FIDI	31:24								
		23:16								
		15:8	FI_DI_RATIO[15:8]							
		7:0	FI_DI_RATIO[7:0]							
0x0244	FLEX_US_NER	31:24								
		23:16								
		15:8								
		7:0	NB_ERRORS[7:0]							
0x0248 ... 0x024B	Reserved									
0x024C	FLEX_US_IF	31:24								
		23:16								
		15:8								
		7:0	IRDA_FILTER[7:0]							
0x0250	FLEX_US_MAN	31:24	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
		23:16					RX_PL[3:0]			
		15:8				TX_MPOL			TX_PP[1:0]	
		7:0					TX_PL[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0254	FLEX_US_LINMR	31:24								
		23:16							SYNCDIS	PDCM
		15:8	DLC[7:0]							
		7:0	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
0x0258	FLEX_US_LINIR	31:24								
		23:16								
		15:8								
		7:0	IDCHR[7:0]							
0x025C	FLEX_US_LINBRR	31:24								
		23:16						LINFP[2:0]		
		15:8	LINCD[15:8]							
		7:0	LINCD[7:0]							
0x0260 ... 0x028F	Reserved									
0x0290	FLEX_US_CMPR	31:24								VAL2[8]
		23:16	VAL2[7:0]							
		15:8		CMPPAR	CMPMODE[1:0]					VAL1[8]
		7:0	VAL1[7:0]							
0x0294 ... 0x029F	Reserved									
0x02A0	FLEX_US_FMR	31:24			RXFTHRES2[5:0]					
		23:16			RXFTHRES[5:0]					
		15:8			TXFTHRES[5:0]					
		7:0	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
0x02A4	FLEX_US_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x02A8	FLEX_US_FIER	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02AC	FLEX_US_FIDR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B0	FLEX_US_FIMR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B4	FLEX_US_FESR	31:24								
		23:16								
		15:8							RXFTHF2	TXFLOCK
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B8 ... 0x02E3	Reserved									
0x02E4	FLEX_US_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x02E8	FLEX_US_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02EC ... 0x03FF	Reserved									
0x0400	FLEX_SPI_CR	31:24	FIFODIS	FIFOEN						LASTXFER
		23:16							RXFCLR	TXFCLR
		15:8				REQCLR				
		7:0	SWRST						SPIDIS	SPIEN
0x0404	FLEX_SPI_MR	31:24	DLYBCS[7:0]							
		23:16					PCS[3:0]			
		15:8				CMPMODE				
		7:0	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
0x0408	FLEX_SPI_RDR	31:24								
		23:16					PCS[3:0]			
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_8)	31:24	RD3[7:0]							
		23:16	RD2[7:0]							
		15:8	RD1[7:0]							
		7:0	RD0[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_16)	31:24	RD1[15:8]							
		23:16	RD1[7:0]							
		15:8	RD0[15:8]							
		7:0	RD0[7:0]							
0x040C	FLEX_SPI_TDR	31:24								LASTXFER
		23:16					PCS[3:0]			
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x040C	FLEX_SPI_TDR (FIFO_MULTI_DATA)	31:24	TD1[15:8]							
		23:16	TD1[7:0]							
		15:8	TD0[15:8]							
		7:0	TD0[7:0]							
0x0410	FLEX_SPI_SR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								SPIENS
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0414	FLEX_SPI_IER	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0418	FLEX_SPI_IDR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x041C	FLEX_SPI_IMR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0420 ... 0x042F	Reserved									
0x0430	FLEX_SPI_CSR0	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0434	FLEX_SPI_CSR1	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0438	FLEX_SPI_CSR2	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x043C	FLEX_SPI_CSR3	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0440	FLEX_SPI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x0444	FLEX_SPI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0448	FLEX_SPI_CMPR	31:24	VAL2[15:8]							
		23:16	VAL2[7:0]							
		15:8	VAL1[15:8]							
		7:0	VAL1[7:0]							
0x044C ... 0x04E3	Reserved									
0x04E4	FLEX_SPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x04E8	FLEX_SPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x04EC ... 0x05FF	Reserved									
0x0600	FLEX_TWI_CR	31:24			FIFODIS	FIFOEN		LOCKCLR		THRCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0600	FLEX_TWI_CR (FIFO_ENABLED)	31:24			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0604	FLEX_TWI_MMR	31:24								NOAP
		23:16		DADR[6:0]						
		15:8		SCLRBL[1:0]		MREAD			IADRSZ[1:0]	
		7:0								
0x0608	FLEX_TWI_SMR	31:24	DATAMEN	SADR3EN	SADR2EN	SADR1EN				
		23:16		SADR[6:0]						
		15:8		MASK[6:0]						
		7:0	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
0x060C	FLEX_TWI_IADR	31:24								
		23:16	IADR[23:16]							
		15:8	IADR[15:8]							
		7:0	IADR[7:0]							
0x0610	FLEX_TWI_CWGR	31:24			HOLD[5:0]					
		23:16				BRSRCCLK		CKDIV[2:0]		
		15:8	CHDIV[7:0]							
		7:0	CLDIV[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0614 ... 0x061F	Reserved									
0x0620	FLEX_TWI_SR	31:24						SR	SDA	SCL
		23:16	LOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
		15:8					EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
0x0620	FLEX_TWI_SR (FIFO_ENABLED)	31:24						SR	SDA	SCL
		23:16	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
		15:8					EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
0x0624	FLEX_TWI_IER	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0628	FLEX_TWI_IDR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x062C	FLEX_TWI_IMR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0630	FLEX_TWI_RHR	31:24								
		23:16								
		15:8				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
		7:0	RXDATA[7:0]							
0x0630	FLEX_TWI_RHR (FIFO_ENABLED)	31:24	RXDATA3[7:0]							
		23:16	RXDATA2[7:0]							
		15:8	RXDATA1[7:0]							
		7:0	RXDATA0[7:0]							
0x0634	FLEX_TWI_THR	31:24								
		23:16								
		15:8								
		7:0	TXDATA[7:0]							
0x0634	FLEX_TWI_THR (FIFO_ENABLED)	31:24	TXDATA3[7:0]							
		23:16	TXDATA2[7:0]							
		15:8	TXDATA1[7:0]							
		7:0	TXDATA0[7:0]							
0x0638	FLEX_TWI_SMBTR	31:24	THMAX[7:0]							
		23:16	TLOWM[7:0]							
		15:8	TLOWS[7:0]							
		7:0					PRESC[3:0]			
0x063C	FLEX_TWI_HSR	31:24								
		23:16								
		15:8								
		7:0	MCODE[7:0]							
0x0640	FLEX_TWI_ACR	31:24							NPEC	NDIR
		23:16	NDATAL[7:0]							
		15:8							PEC	DIR
		7:0	DATAL[7:0]							
0x0644	FLEX_TWI_FILTR	31:24								
		23:16								
		15:8						THRES[2:0]		
		7:0							PADFEN	FILT
0x0648	FLEX_TWI_HSCWGR	31:24								
		23:16						HSCKDIV[2:0]		
		15:8	HSCHDIV[7:0]							
		7:0	HSCLDIV[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x064C	FLEX_TWI_SWMR	31:24	DATAM[7:0]							
		23:16		SADR3[6:0]						
		15:8		SADR2[6:0]						
		7:0		SADR1[6:0]						
0x0650	FLEX_TWI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x0654	FLEX_TWI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0658 ... 0x065F	Reserved									
0x0660	FLEX_TWI_FSR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0664	FLEX_TWI_FIER	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0668	FLEX_TWI_FIDR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x066C	FLEX_TWI_FIMR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0670 ... 0x06E3	Reserved									
0x06E4	FLEX_TWI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x06E8	FLEX_TWI_WPSR	31:24	WPVSRC[23:16]							
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

63.10.1 FLEXCOM Mode Register

Name: FLEX_MR
Offset: 0x000
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OPMODE[1:0]	
Access							R/W	R/W
Reset							0	1

Bits 1:0 – OPMODE[1:0] FLEXCOM Operating Mode

Value	Name	Description
0	NO_COM	No communication
1	USART	All UART-related protocols are selected (RS232, RS485, IrDA, ISO7816, LIN,) SPI/TWI-related registers are not accessible and have no impact on IOs.
2	SPI	SPI operating mode is selected. USART/TWI related registers are not accessible and have no impact on IOs.
3	TWI	All TWI-related protocols are selected (TWI, SMBus). USART/SPI-related registers are not accessible and have no impact on IOs.

63.10.2 FLEXCOM Receive Holding Register

Name: FLEX_RHR
Offset: 0x010
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXDATA[15:0] Receive Data
This register is a mirror of:
• USART Receive Holding Register (FLEX_US_RHR) if FLEX_MR.OPMODE field equals 1
• SPI Receive Data Register (FLEX_SPI_RDR) if FLEX_MR.OPMODE field equals 2
• TWI Transmit Holding Register (FLEX_TWI_RHR) if FLEX_MR.OPMODE field equals 3

63.10.3 FLEXCOM Transmit Holding Register

Name: FLEX_THR
Offset: 0x020
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXDATA[15:0] Transmit Data
This register is a mirror of:
- USART Transmit Holding Register (FLEX_US_THR) if FLEX_MR.OPMODE field equals 1
- SPI Transmit Data Register (FLEX_SPI_TDR) if FLEX_MR.OPMODE field equals 2
- TWI Transmit Holding Register (FLEX_TWI_THR) if FLEX_MR.OPMODE field equals 3

63.10.4 USART Control Register

Name: FLEX_US_CR
Offset: 0x200
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
Access	W	W		W		W	W	W
Reset	–	–		–		–	–	–

Bit	23	22	21	20	19	18	17	16
			LINWKUP	LINABT	RTSDIS	RTSEN		
Access			W	W	W	W		
Reset			–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs.

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs.

Bit 28 – REQCLR Request to Clear the Comparison Trigger
Asynchronous partial wakeup enabled:
0: No effect.
1: Clears the potential clock request currently issued by USART, thus the potential system wakeup is cancelled.
Asynchronous partial wakeup disabled:
0: No effect.
1: Restarts the comparison trigger to enable FLEX_US_RHR loading.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 21 – LINWKUP Send LIN Wakeup Signal

Value	Description
0	No effect:
1	Sends a wakeup signal on the LIN bus.

Bit 20 – LINABT Abort LIN Transmission

Value	Description
0	No effect.
1	Aborts the current LIN transmission.

Bit 19 – RTSDIS Request to Send Disable

Value	Description
0	No effect.
1	Drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 0.

Bit 18 – RTSEN Request to Send Enable

Value	Description
0	No effect.
1	Drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 0.

Bit 15 – RETTO Start Timeout Immediately

Value	Description
0	No effect
1	Immediately restarts timeout period.

Bit 14 – RSTNACK Reset Non Acknowledge

Value	Description
0	No effect
1	Resets FLEX_US_CSR.NACK.

Bit 13 – RSTIT Reset Iterations

Value	Description
0	No effect.
1	Resets FLEX_US_CSR.ITER. No effect if the ISO7816 is not enabled.

Bit 12 – SENDA Send Address

Value	Description
0	No effect.
1	In Multidrop mode only, the next character written to FLEX_US_THR is sent with the address bit set.

Bit 11 – STTTO Clear TIMEOUT Flag and Start Timeout After Next Character Received

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the timeout counter. Immediately disables a timeout period in progress. Resets the FLEX_US_CSR.TIMEOUT status bit.

Bit 10 – STPBRK Stop Break

Value	Description
0	No effect.

Value	Description
1	Stops transmission of the break after a minimum of one character length and transmits a high level during 12-bit periods. No effect if no break is being transmitted.

Bit 9 – STTBRK Start Break

Value	Description
0	No effect.
1	Starts transmission of a break after the characters present in FLEX_US_THR and the Transmit Shift Register have been transmitted. No effect if a break is already being transmitted.

Bit 8 – RSTSTA Reset Status Bits

Value	Description
0	No effect.
1	Resets the PARE, FRAME, OVRE, MANE, LINBE, LINISFE, LINIPE, LINCE, LINSNRE, LINSTE, LINHTE, LINID, LINTC, LINBK, CMP and RXBRK in FLEX_US_CSR status bits, as well as the TXFEF, TXFFF, TXFTHF, RXFEF, RXFFF, RXFTHF, TXFPTEF, RXFPTEF in FLEX_US_FESR status bits.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	Disables the transmitter.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	Enables the transmitter if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	Disables the receiver.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	Enables the receiver, if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

Value	Description
0	No effect.
1	Resets the transmitter.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	Resets the receiver.

63.10.5 USART Mode Register

Name: FLEX_US_MR
Offset: 0x204
Reset: 0xC0000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	–	–	–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit 31 – ONEBIT Start Frame Delimiter Selector

Value	Description
0	Start frame delimiter is COMMAND or DATA SYNC.
1	Start frame delimiter is one bit.

Bit 30 – MODSYNC Manchester Synchronization Mode

Value	Description
0	The Manchester start bit is a 0 to 1 transition
1	The Manchester start bit is a 1 to 0 transition.

Bit 29 – MAN Manchester Encoder/Decoder Enable

Value	Description
0	Manchester encoder/decoder are disabled.
1	Manchester encoder/decoder are enabled.

Bit 28 – FILTER Receive Line Filter

Value	Description
0	The USART does not filter the receive line.
1	The USART filters the receive line using a three-sample filter (1/16-bit clock) (2 over 3 majority).

Bits 26:24 – MAX_ITERATION[2:0] Maximum Number of Automatic Iterations

Value	Description
0–7	Defines the maximum number of iterations in mode ISO7816, protocol T = 0.

Bit 23 – INVDATA Inverted Data

Value	Description
0	The data field transmitted on TXD line is the same as the one written in FLEX_US_THR or the content read in FLEX_US_RHR is the same as RXD line. Normal mode of operation.

Value	Description
1	The data field transmitted on TXD line is inverted (voltage polarity only) compared to the value written in FLEX_US_THR or the content read in FLEX_US_RHR is inverted compared to what is received on RXD line (or ISO7816 IO line). Inverted mode of operation, useful for contactless card application. To be used with configuration bit MSBF.

Bit 22 – VAR_SYNC Variable Synchronization of Command/Data Sync Start Frame Delimiter

Value	Description
0	User defined configuration of command or data sync field depending on MODSYNC value.
1	The sync field is updated when a character is written into FLEX_US_THR.

Bit 21 – DSNACK Disable Successive NACK

The MAX_ITERATION field must be cleared if DSNACK is cleared.

Value	Description
0	NACK is sent on the ISO line as soon as a parity error occurs in the received character (unless INACK is set).
1	Successive parity errors are counted up to the value specified in the MAX_ITERATION field. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line. The flag ITER is asserted.

Bit 20 – INACK Inhibit Non Acknowledge

Value	Description
0	The NACK is generated.
1	The NACK is not generated.

Bit 19 – OVER Oversampling Mode

Value	Description
0	16x Oversampling.
1	8x Oversampling.

Bit 18 – CLKO Clock Output Select

Value	Description
0	The USART does not drive the SCK pin (Synchronous Client mode or Asynchronous mode with external baud rate clock source).
1	The USART drives the SCK pin if USCLKS does not select the external clock SCK (USART Synchronous Host mode).

Bit 17 – MODE9 9-bit Character Length

Value	Description
0	CHRL defines character length.
1	9-bit character length.

Bit 16 – MSBF Bit Order

Value	Description
0	Least significant bit is sent/received first.
1	Most significant bit is sent/received first.

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic Echo. Receiver input is connected to the TXD pin.
2	LOCAL_LOOPBACK	Local Loopback. Transmitter output is connected to the Receiver Input.
3	REMOTE_LOOPBACK	Remote Loopback. RXD pin is internally connected to the TXD pin.

Bits 13:12 – NBSTOP[1:0] Number of Stop Bits

Value	Name	Description
0	1_BIT	1 stop bit

Value	Name	Description
1	1_5_BIT	1.5 stop bit (SYNC = 0) or reserved (SYNC = 1)
2	2_BIT	2 stop bits

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Parity forced to 0 (Space)
3	MARK	Parity forced to 1 (Mark)
4	NO	No parity
6	MULTIDROP	Multidrop mode

Bit 8 – SYNC Synchronous Mode Select

Value	Description
0	USART operates in Asynchronous mode (UART).
1	USART operates in Synchronous mode.

Bits 7:6 – CHRL[1:0] Character Length

Value	Name	Description
0	5_BIT	Character length is 5 bits
1	6_BIT	Character length is 6 bits
2	7_BIT	Character length is 7 bits
3	8_BIT	Character length is 8 bits

Bits 5:4 – USCLKS[1:0] Clock Selection

Value	Name	Description
0	MCK	Peripheral clock is selected
–	–	Reserved
2	GCLK	PMC generic clock is selected. If the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.
3	SCK	External pin SCK is selected

Bits 3:0 – USART_MODE[3:0] USART Mode of Operation
Values not listed in the table below should be considered 'reserved'.

Value	Name	Description
0x0	NORMAL	Normal mode
0x1	RS485	RS485
0x2	HW_HANDSHAKING	Hardware handshaking
0x4	IS07816_T_0	IS07816 Protocol: T = 0
0x6	IS07816_T_1	IS07816 Protocol: T = 1
0x8	IRDA	IrDA
0xA	LIN_MASTER	LIN Host mode
0xB	LIN_SLAVE	LIN Client mode
0xC	DATA16BIT_MASTER	16-bit data host
0xD	DATA16BIT_SLAVE	16-bit data client

63.10.6 USART Interrupt Enable Register

Name: FLEX_US_IER
Offset: 0x208
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Enable Register (LIN_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			W			W	W	W
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	W	W	W			W	W	W
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error Interrupt Enable

Bit 22 – CMP Comparison Interrupt Enable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Enable

Bit 13 – NACK Non Acknowledge Interrupt Enable

Bit 10 – ITER Max number of Repetitions Reached Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 2 – RXBRK Receiver Break Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

63.10.7 USART Interrupt Enable Register (LIN_MODE)

Name: FLEX_US_IER (LIN_MODE)
Offset: 0x208
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Enable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Enable

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Enable

Bit 28 – LINCE LIN Checksum Error Interrupt Enable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Enable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Enable

Bit 25 – LINBE LIN Bus Error Interrupt Enable

Bit 15 – LINTC LIN Transfer Completed Interrupt Enable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Enable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

63.10.8 USART Interrupt Disable Register

Name: FLEX_US_IDR
Offset: 0x20C
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Disable Register (LIN_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			W			W	W	W
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	W	W	W			W	W	W
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error Interrupt Disable

Bit 22 – CMP Comparison Interrupt Disable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Disable

Bit 13 – NACK Non Acknowledge Interrupt Disable

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 2 – RXBRK Receiver Break Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

63.10.9 USART Interrupt Disable Register (LIN_MODE)

Name: FLEX_US_IDR (LIN_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Disable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Disable

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Disable

Bit 28 – LINCE LIN Checksum Error Interrupt Disable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Disable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Disable

Bit 25 – LINBE LIN Bus Error Interrupt Disable

Bit 15 – LINTC LIN Transfer Completed Interrupt Disable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Disable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

63.10.10 USART Interrupt Mask Register

Name: FLEX_US_IMR
Offset: 0x210
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Interrupt Mask Register (LIN_MODE).

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								0
Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		R			R			
Reset		0			0			
Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			R			R	R	R
Reset			0			0	0	0
Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	R	R	R			R	R	R
Reset	0	0	0			0	0	0

Bit 24 – MANE Manchester Error Interrupt Mask

Bit 22 – CMP Comparison Interrupt Mask

Bit 19 – CTSIC Clear to Send Input Change Interrupt Mask

Bit 13 – NACK Non Acknowledge Interrupt Mask

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 2 – RXBRK Receiver Break Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

63.10.11 USART Interrupt Mask Register (LIN_MODE)

Name: FLEX_US_IMR (LIN_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Mask

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Mask

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Mask

Bit 28 – LINCE LIN Checksum Error Interrupt Mask

Bit 27 – LINIPE LIN Identifier Parity Interrupt Mask

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Mask

Bit 25 – LINBE LIN Bus Error Interrupt Mask

Bit 15 – LINTC LIN Transfer Completed Interrupt Mask

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Mask

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

63.10.12 USART Channel Status Register

Name: FLEX_US_CSR
Offset: 0x214
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Channel Status Register (LIN_MODE).

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								–

Bit	23	22	21	20	19	18	17	16
	CTS	CMP			CTSIC			
Access	R	R			R			
Reset	–	–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			R			R	R	R
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	R	R	R			R	R	R
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error

Value	Description
0	No Manchester error has been detected since the last RSTSTA command was issued.
1	At least one Manchester error has been detected since the last RSTSTA command was issued.

Bit 23 – CTS Image of CTS Input

Value	Description
0	CTS input is driven low.
1	CTS input is driven high.

Bit 22 – CMP Comparison Status

Value	Description
0	No received character matched the comparison criteria programmed in VAL1, VAL2 fields and CMPPAR bit in since the last RSTSTA command was issued.
1	A received character matched the comparison criteria since the last RSTSTA command was issued.

Bit 19 – CTSIC Clear to Send Input Change Flag

Value	Description
0	No input change has been detected on the CTS pin since the last read of FLEX_US_CSR.
1	At least one input change has been detected on the CTS pin since the last read of FLEX_US_CSR.

Bit 13 – NACK Non Acknowledge Interrupt

Value	Description
0	Non acknowledge has not been detected since the last RSTNACK.
1	At least one non acknowledge has been detected since the last RSTNACK.

Bit 10 – ITER Max Number of Repetitions Reached

Value	Description
0	Maximum number of repetitions has not been reached since the last RSTIT command was issued.
1	Maximum number of repetitions has been reached since the last RSTIT command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 2 – RXBRK Break Received/End of Break

Value	Description
0	No break received or end of break detected since the last RSTSTA command was issued.
1	Break received or end of break detected since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)
When FIFOs are disabled:
0: A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or an STTBRK command has been requested, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1: There is no character in FLEX_US_THR.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFO enabled is illustrated in 63.7.11.5. TXEMPTY, TXRDY and RXRDY Behavior.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)
When FIFOs are disabled:
0: No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1: At least one complete character has been received and FLEX_US_RHR has not yet been read.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read
1: At least one unread data is in the Receive FIFO
RXRDY behavior with FIFO enabled is illustrated in 63.7.11.5. TXEMPTY, TXRDY and RXRDY Behavior.

63.10.13 USART Channel Status Register (LIN_MODE)

Name: FLEX_US_CSR (LIN_MODE)
Offset: 0x214
Reset: –
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
	LINBLS							
Access	R							
Reset	–							

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error

Value	Description
0	No LIN header timeout error has been detected since the last RSTSTA command was issued.
1	A LIN header timeout error has been detected since the last RSTSTA command was issued.

Bit 30 – LINSTE LIN Synch Tolerance Error

Value	Description
0	No LIN synch tolerance error has been detected since the last RSTSTA command was issued.
1	A LIN synch tolerance error has been detected since the last RSTSTA command was issued.

Bit 29 – LINSNRE LIN Client Not Responding Error

Value	Description
0	No LIN client not responding error has been detected since the last RSTSTA command was issued.
1	A LIN client not responding error has been detected since the last RSTSTA command was issued.

Bit 28 – LINCE LIN Checksum Error

Value	Description
0	No LIN checksum error has been detected since the last RSTSTA command was issued.
1	A LIN checksum error has been detected since the last RSTSTA command was issued.

Bit 27 – LINIPE LIN Identifier Parity Error

Value	Description
0	No LIN identifier parity error has been detected since the last RSTSTA command was issued.
1	A LIN identifier parity error has been detected since the last RSTSTA command was issued.

Bit 26 – LINISFE LIN Inconsistent Synch Field Error

Value	Description
0	No LIN inconsistent synch field error has been detected since the last RSTSTA
1	The USART is configured as a client node and a LIN Inconsistent synch field error has been detected since the last RSTSTA command was issued.

Bit 25 – LINBE LIN Bit Error

Value	Description
0	No bit error has been detected since the last RSTSTA command was issued.
1	A bit error has been detected since the last RSTSTA command was issued.

Bit 23 – LINBLS LIN Bus Line Status

Value	Description
0	LIN bus line is set to 0.
1	LIN bus line is set to 1.

Bit 15 – LINTC LIN Transfer Completed

Value	Description
0	The USART is idle or a LIN transfer is ongoing.
1	A LIN transfer has been completed since the last RSTSTA command was issued.

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received
If USART operates in LIN Host mode (USART_MODE = 0xA):
0: No LIN identifier has been sent since the last RSTSTA command was issued.
1: At least one LIN identifier has been sent since the last RSTSTA command was issued.
If USART operates in LIN Client mode (USART_MODE = 0xB):
0: No LIN identifier has been received since the last RSTSTA command was issued.
1: At least one LIN identifier has been received since the last RSTSTA.

Bit 13 – LINBK LIN Break Sent or LIN Break Received
Applicable if USART operates in LIN Host mode (USART_MODE = 0xA):
0: No LIN break has been sent since the last RSTSTA command was issued.
1: At least one LIN break has been sent since the last RSTSTA.
If USART operates in LIN Client mode (USART_MODE = 0xB):
0: No LIN break has received sent since the last RSTSTA command was issued.
1: At least one LIN break has been received since the last RSTSTA command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last start timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last start timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)

Value	Description
0	A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1	There is no character in FLEX_US_THR.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)

Value	Description
0	No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1	At least one complete character has been received and FLEX_US_RHR has not yet been read.

63.10.14 USART Receive Holding Register

Name:	FLEX_US_RHR
Offset:	0x218
Reset:	0x00000000
Property:	Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), a byte access on FLEX_SPI_TDR reads one byte (FLEX_US_RHR.MODE9=0), see 63.7.11.6. FIFO Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXSYNH							RXCHR[8]
Access	R							R
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – RXSYNH Received Sync

Value	Description
0	Last character received is a data.
1	Last character received is a command.

Bits 8:0 – RXCHR[8:0] Received Character
Last character received if RXRDY is set.

63.10.15 USART Receive Holding Register (FIFO Multi Data)

Name: FLEX_US_RHR (FIFO_MULTI_DATA)
Offset: 0x218
Reset: 0x00000000
Property: Read-only

To read multi-data in a single access, the FIFO must be enabled (FLEX_US_CR.FIFOEN=1) and FLEX_US_MR.MODE9=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2, 4), see 63.7.11.7. FIFO Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RXCHR3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RXCHR2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXCHR1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RXCHR0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RXCHRx Received Character
First unread character in the Receive FIFO if RXRDY is set.

63.10.16 USART Transmit Holding Register

Name: FLEX_US_THR
Offset: 0x21C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), a byte access on FLEX_SPI_TDR writes one byte (FLEX_US_MR.MODE9=0), see 63.7.11.6. FIFO Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXSYNH							TXCHR[8]
Access	W							W
Reset	–							–

Bit	7	6	5	4	3	2	1	0
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 15 – TXSYNH Sync Field to be Transmitted

Value	Description
0	The next character sent is encoded as a data. Start frame delimiter is DATA SYNC.
1	The next character sent is encoded as a command. Start frame delimiter is COMMAND SYNC.

Bits 8:0 – TXCHR[8:0] Character to be Transmitted
The next character to be transmitted after the current character if TXRDY is not set.

63.10.17 USART Transmit Holding Register (FIFO Multi Data)

Name: FLEX_US_THR (FIFO_MULTI_DATA)
Offset: 0x21C
Reset: –
Property: Write-only

To write multi-data in a single access, the FIFO must be enabled (FLEX_US_CR.FIFOEN=1) and FLEX_US_MR.MODE9=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see 63.7.11.7. FIFO Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TXCHR3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	TXCHR2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	TXCHR1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	TXCHR0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:7, 8:15, 16:23, 24:31 – TXCHRx Character to be Transmitted
Next character to be transmitted.

63.10.18 USART Baud Rate Generator Register

Name: FLEX_US_BRGR
Offset: 0x220
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – FP[2:0] Fractional Part

WARNING When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by “selected clock” period from time to time. The duty cycle depends on the value of the CD field.

Value	Description
0	Fractional divider is disabled.
1–7	Baud rate resolution, defined by FP × 1/8.

Bits 15:0 – CD[15:0] Clock Divider

CD	USART_MODE ≠ ISO7816			USART_MODE = ISO7816
	SYNC = 0		SYNC = 1	
	OVER = 0	OVER = 1		
0	Baud Rate Clock Disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

63.10.19 USART Receiver Timeout Register

Name: FLEX_US_RTOR
Offset: 0x224
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TO[16]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16:0 – TO[16:0] Timeout Value
The TO field size is limited to 8 bits if the ISO7816 logic is not implemented on some instances of FLEXCOM. The ISO7816 logic is implemented if it is possible to write FLEX_US_MR.MAX_ITERATIONS=1 (a read operation must be performed after the write operation to check that MAX_ITERATIONS equals 1).

Value	Description
0	The receiver timeout is disabled.
1-131071	The receiver timeout is enabled and the timeout delay is TO × bit period.

63.10.20 USART Transmitter Timeguard Register

Name: FLEX_US_TTGR
Offset: 0x228
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	TG[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TG[7:0] Timeguard Value

Value	Description
0	The transmitter timeguard is disabled.
1-255	The transmitter timeguard is enabled and TG is timeguard delay / bit period.

63.10.21 USART FI DI RATIO Register

Name: FLEX_US_FIDI
Offset: 0x240
Reset: 0x174
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FI_DI_RATIO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	FI_DI_RATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	0	0

Bits 15:0 – FI_DI_RATIO[15:0] FI Over DI Ratio Value

Value	Description
0	If ISO7816 mode is selected, the baud rate generator generates no signal.
1-2	Do not use.
3-65535	If ISO7816 mode is selected, the baud rate is the clock provided on SCK divided by FI_DI_RATIO.

63.10.22 USART Number of Errors Register

Name: FLEX_US_NER
Offset: 0x244
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0x4 or 0x6 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	NB_ERRORS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – NB_ERRORS[7:0] Number of Errors
Total number of errors that occurred during an ISO7816 transfer. This register automatically clears when read.

63.10.23 USART IrDA FILTER Register

Name: FLEX_US_IF
Offset: 0x24C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x8 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IRDA_FILTER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IRDA_FILTER[7:0] IrDA Filter
The IRDA_FILTER value must be defined to meet the following criteria:
$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$

63.10.24 USART Manchester Configuration Register

Name: FLEX_US_MAN
Offset: 0x250
Reset: 0xB0011004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	1	0	1	1			0	0

Bit	23	22	21	20	19	18	17	16
					RX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bit	15	14	13	12	11	10	9	8
				TX_MPOL			TX_PP[1:0]	
Access				R/W			R/W	R/W
Reset				1			0	0

Bit	7	6	5	4	3	2	1	0
					TX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit 31 – RXIDLEV Receiver Idle Value

Value	Description
0	Receiver line idle value is 0.
1	Receiver line idle value is 1.

Bit 30 – DRIFT Drift Compensation

Value	Description
0	The USART cannot recover from an important clock drift.
1	The USART can recover from clock drift. The 16X Clock mode must be enabled.

Bit 29 – ONE Must Be Set to 1
Bit 29 must always be set to 1 when programming the FLEX_US_MAN register.

Bit 28 – RX_MPOL Receiver Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 25:24 – RX_PP[1:0] Receiver Preamble Pattern detected
The following values assume that RX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s.
1	ALL_ZERO	The preamble is composed of ‘0’s.
2	ZERO_ONE	The preamble is composed of ‘01’s.
3	ONE_ZERO	The preamble is composed of ‘10’s.

Bits 19:16 – RX_PL[3:0] Receiver Preamble Length

Value	Description
0	The receiver preamble pattern detection is disabled.
1–15	The detected preamble length is RX_PL × Bit Period.

Bit 12 – TX_MPOL Transmitter Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 9:8 – TX_PP[1:0] Transmitter Preamble Pattern

The following values assume that TX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s.
1	ALL_ZERO	The preamble is composed of ‘0’s.
2	ZERO_ONE	The preamble is composed of ‘01’s.
3	ONE_ZERO	The preamble is composed of ‘10’s.

Bits 3:0 – TX_PL[3:0] Transmitter Preamble Length

Value	Description
0	The transmitter preamble pattern generation is disabled.
1–15	The preamble length is TX_PL × Bit Period.

63.10.25 USART LIN Mode Register

Name: FLEX_US_LINMR
Offset: 0x254
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SYNCDIS	PDCM
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DLC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – SYNCDIS Synchronization Disable

Value	Description
0	The synchronization procedure is performed in LIN client node configuration.
1	The synchronization procedure is not performed in LIN client node configuration.

Bit 16 – PDCM DMAC Mode

Value	Description
0	The LIN mode register FLEX_US_LINMR is not written by the DMAC.
1	The LIN mode register FLEX_US_LINMR (excepting that flag) is written by the DMAC.

Bits 15:8 – DLC[7:0] Data Length Control

Value	Description
0-255	Defines the response data length if DLM = 0, in that case the response data length is equal to DLC+1 bytes.

Bit 7 – WKUPTYP Wake-up Signal Type

Value	Description
0	Setting the LINWKUP bit in the control register sends a LIN 2.0 wake-up signal.
1	Setting the LINWKUP bit in the control register sends a LIN 1.3 wake-up signal.

Bit 6 – FSDIS Frame Slot Mode Disable

Value	Description
0	The Frame Slot mode is enabled.
1	The Frame Slot mode is disabled.

Bit 5 – DLM Data Length Mode

Value	Description
0	The response data length is defined by the DLC field of this register.

Value	Description
1	The response data length is defined by the bits 5 and 6 of the identifier (FLEX_US_LINIR.IDCHR).

Bit 4 – CHKTYP Checksum Type

Value	Description
0	LIN 2.0 “enhanced” checksum
1	LIN 1.3 “classic” checksum

Bit 3 – CHKDIS Checksum Disable

Value	Description
0	In host node configuration, the checksum is computed and sent automatically. In client node configuration, the checksum is checked automatically.
1	Whatever the node configuration is, the checksum is not computed/sent and it is not checked.

Bit 2 – PARDIS Parity Disable

Value	Description
0	In host node configuration, the identifier parity is computed and sent automatically. In host node and client node configuration, the parity is checked automatically.
1	Whatever the node configuration is, the Identifier parity is not computed/sent and it is not checked.

Bits 1:0 – NACT[1:0] LIN Node Action

Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	PUBLISH	The USART transmits the response.
1	SUBSCRIBE	The USART receives the response.
2	IGNORE	The USART does not transmit and does not receive the response.

63.10.26 USART LIN Identifier Register

Name: FLEX_US_LINIR
Offset: 0x258
Reset: 0x00000000
Property: Read/Write

Write is possible only in LIN host node configuration.

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	IDCHR[7:0]							
Access	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IDCHR[7:0] Identifier Character
If USART_MODE = 0xA (host node configuration):

- IDCHR is Read/Write and its value is the identifier character to be transmitted.

If USART_MODE = 0xB (client node configuration):

- IDCHR is Read-only and its value is the last identifier character that has been received.

63.10.27 USART LIN Baud Rate Register

Name: FLEX_US_LINBRR
Offset: 0x25C
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Returns the baud rate value after the synchronization process completion.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						LINFP[2:0]		
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	LINCD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LINCD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – LINFP[2:0] Fractional Part after Synchronization

Bits 15:0 – LINCD[15:0] Clock Divider after Synchronization

63.10.28 USART Comparison Register

Name: FLEX_US_CMPR
Offset: 0x290
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								VAL2[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		CMPPAR	CMPMODE[1:0]					VAL1[8]
Access		R/W	R/W	R/W				R/W
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24:16 – VAL2[8:0] Second Comparison Value for Received Character

Value	Description
`0-511`	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_US_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the UART requests a system wakeup if condition is met.

Bit 14 – CMPPAR Compare Parity

Value	Description
`0`	The parity is not checked and a bad parity cannot prevent from waking up the system.
`1`	The parity is checked and a matching condition on data can be cancelled by an error on parity bit, so no wakeup is performed.

Bits 13:12 – CMPMODE[1:0] Comparison Mode

Value	Name	Description
`0`	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
`1`	START_CONDITION	Comparison condition must be met to start reception.
`2`	FILTER	Comparison must be met to receive the current data only

Bits 8:0 – VAL1[8:0] First Comparison Value for Received Character

Value	Description
`0-511`	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_US_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the UART requests a system wakeup if the condition is met.

63.10.29 USART FIFO Mode Register

Name: FLEX_US_FMR
Offset: 0x2A0
Reset: 0x00000000
Property: Read/Write

This register reads ‘0’ if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
			RXFTHRES2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bits 29:24 – RXFTHRES2[5:0] Receive FIFO Threshold 2

Value	Description
0-32	Defines the Receive FIFO threshold 2 value (number of bytes). The FLEX_US_FESR.RXFTHF2 flag will be set when Receive FIFO goes from “above” threshold state to “equal to or below” threshold state.

Bits 21:16 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-32	Defines the Receive FIFO threshold value (number of bytes). The FLEX_US_FESR.RXFTHF flag will be set when Receive FIFO goes from “below” threshold state to “equal to or above” threshold state.

Bits 13:8 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-32	Defines the Transmit FIFO threshold value (number of bytes). The FLEX_US_FESR.TXFTHF flag will be set when Transmit FIFO goes from “above” threshold state to “equal to or below” threshold state.

Bit 7 – FRTSC FIFO RTS Pin Control enable (Hardware Handshaking mode only)
See Hardware Handshaking for details.

Value	Description
0	RTS pin is not controlled by Receive FIFO thresholds.
1	RTS pin is controlled by Receive FIFO thresholds.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.RXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RXRDY will be at level ‘1’ when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as a byte.
1	TWO_DATA	RXRDY will be at level ‘1’ when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA (chunk size=1 and halfword access). If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	RXRDY will be at level ‘1’ when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level ‘1’ when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as a byte.
1	TWO_DATA	TXRDY will be at level ‘1’ when at least two data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	TXRDY will be at level ‘1’ when at least four data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA (chunk size=1 and word access). If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

63.10.30 USART FIFO Level Register

Name: FLEX_US_FLR
Offset: 0x2A4
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–32	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–32	Indicates the number of data in the Transmit FIFO.

63.10.31 USART FIFO Interrupt Enable Register

Name: FLEX_US_FIER
Offset: 0x2A8
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Enable

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

63.10.32 USART FIFO Interrupt Disable Register

Name: FLEX_US_FIDR
Offset: 0x2AC
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Disable

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

63.10.33 USART FIFO Interrupt Mask Register

Name: FLEX_US_FIMR
Offset: 0x2B0
Reset: 0x00000000
Property: Read-only

This register reads ‘0’ if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							RXFTHF2	
Access							R	
Reset							0	
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Mask

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

63.10.34 USART FIFO Event Status Register

Name: FLEX_US_FESR
Offset: 0x2B4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	TXFLOCK
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 Receive FIFO Threshold Flag 2 (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of unread data in Receive FIFO is above RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES2 threshold since the last RSTSTA command was issued.

Bit 8 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See 63.7.11.9. FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See 63.7.11.9. FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last RSTSTA command was issued.

Bit 4 – RXFFF Receive FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last RSTSTA command was issued.

Bit 3 – RXFEF Receive FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last RSTSTA command was issued.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last RSTSTA command was issued.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last RSTSTA command was issued.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last RSTSTA command was issued.

63.10.35 USART Write Protection Mode Register

Name: FLEX_US_WPMR
Offset: 0x2E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x555341	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 0 – WPEN Write Protection Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

63.10.36 USART Write Protection Status Register

Name: FLEX_US_WPSR
Offset: 0x2E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of FLEX_US_WPSR.
1	A write protection violation has occurred since the last read of FLEX_US_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

63.10.37 SPI Control Register

Name: FLEX_SPI_CR
Offset: 0x400
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN						LASTXFER
Access	W	W						W
Reset	–	–						–

Bit	23	22	21	20	19	18	17	16
							RXFCLR	TXFCLR
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
				REQCLR				
Access				W				
Reset				–				

Bit	7	6	5	4	3	2	1	0
	SWRST						SPIDIS	SPIEN
Access	W						W	W
Reset	–						–	–

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs

Bit 24 – LASTXFER Last Transfer
See Peripheral Selection for more details.

Value	Description
0	No effect.
1	The current NPCS will be de-asserted after the character written in TD has been transferred. When CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bit 17 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 16 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 12 – REQCLR Request to Clear the Comparison Trigger
Asynchronous partial wakeup enabled:
0: No effect.
1: Clears the potential clock request currently issued by SPI, thus the potential system wakeup is cancelled.
Asynchronous partial wakeup disabled:
0: No effect.
1: Restarts the comparison trigger to enable FLEX_SPI_RDR loading.

Bit 7 – SWRST SPI Software Reset
The SPI is in Client mode after software reset.

Value	Description
0	No effect.
1	Resets the SPI. A software-triggered hardware reset of the SPI interface is performed.

Bit 1 – SPIDIS SPI Disable
If a transfer is in progress when SPIDIS is set, the SPI completes the transmission of the shifter register and does not start any new transfer, even if the FLEX_US_THR is loaded.
All pins are set to Input mode after completion of the transmission in progress, if any.

Value	Description
0	No effect.
1	Disables the SPI.

Bit 0 – SPIEN SPI Enable

Value	Description
0	No effect.
1	Enables the SPI to transfer and receive data.

63.10.38 SPI Mode Register

Name: FLEX_SPI_MR
Offset: 0x404
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYBCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				CMPMODE				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:24 – DLYBCS[7:0] Delay Between Chip Selects
This field defines the delay between the inactivation and the activation of NPCS. The DLYBCS time ensures chip selects do not overlap and solves bus contentions in case of peripherals having long data float times.
If DLYBCS is ≤ 6, six peripheral clock periods are inserted by default.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCS = Delay Between Chip Selects × $f_{peripheral\ clock}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCS = Delay Between Chip Selects × f_{GCLK}

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if fixed peripheral select is active (PS = 0).
If PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If PCSDEC = 1:
NPCS[3:0] output signals = PCS

Bit 12 – CMPMODE Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception of all incoming characters until REQCLR is set.

Bit 7 – LLB Local Loopback Enable
LLB controls the local loopback on the data shift register for testing in Host mode only (MISO is internally connected on MOSI).

Value	Description
0	Local loopback path disabled.
1	Local loopback path enabled.

Bit 5 – WDRBT Wait Data Read Before Transfer

Value	Description
0	No Effect. In Host mode, a transfer can be initiated regardless of the FLEX_SPI_RDR state.
1	In Host mode, a transfer can start only if FLEX_SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 4 – MODFDIS Mode Fault Detection

Value	Description
0	Mode fault detection is enabled.
1	Mode fault detection is disabled.

Bit 3 – BRSRCCLK Bit Rate Source Clock
If the bit BRSRCCLK = 1, the FLEX_US_CSRx.SCBR field must be programmed with a value greater than 1.

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bit 2 – PCSDEC Chip Select Decode
When PCSDEC equals one, up to 15 Chip Select signals can be generated with the four NPCS lines using an external 4- to 16-bit decoder. The Chip Select registers define the characteristics of the 15 chip selects, with the following rules:
FLEX_SPI_CSR0 defines peripheral chip select signals 0 to 3.
FLEX_SPI_CSR1 defines peripheral chip select signals 4 to 7.
FLEX_SPI_CSR2 defines peripheral chip select signals 8 to 11.
FLEX_SPI_CSR3 defines peripheral chip select signals 12 to 14.

Value	Description
0	The chip selects are directly connected to a peripheral device.
1	The four NPCS chip select lines are connected to a 4- to 16-bit decoder.

Bit 1 – PS Peripheral Select

Value	Description
0	Fixed Peripheral Select
1	Variable Peripheral Select

Bit 0 – MSTR Host/Client Mode

Value	Description
0	SPI is in Client mode.
1	SPI is in Host mode.

63.10.39 SPI Receive Data Register

Name: FLEX_SPI_RDR
Offset: 0x408
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.RXRDYM = 0, see SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – PCS[3:0] Peripheral Chip Select
In Host mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits are read as zero.
Note: When using Variable Peripheral Select mode (FLEX_SPI_MR.PS = 1), it is mandatory to set the FLEX_SPI_MR.WDRBT bit to 1 if the PCS field must be processed in FLEX_SPI_RDR.

Bits 15:0 – RD[15:0] Receive Data
Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

63.10.40 SPI Receive Data Register (FIFO Multiple Data, 8-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_8)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see SPI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RD3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RD2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

63.10.41 SPI Receive Data Register (FIFO Multiple Data, 16-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_16)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1 or 2), see SPI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RD1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD0[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:15, 16:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

63.10.42 SPI Transmit Data Register

Name: FLEX_SPI_TDR
Offset: 0x40C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN=1), a byte/halfword access on FLEX_SPI_TDR writes one byte/halfword, see 63.8.7.6. SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – LASTXFER Last Transfer
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).

Value	Description
0	No effect.
1	The current NPCS is de-asserted after the transfer of the character written in TD. When FLEX_SPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).
If FLEX_SPI_MR.PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If FLEX_SPI_MR.PCSDEC = 1:
NPCS[3:0] output signals = PCS

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the SPI Interface is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

63.10.43 SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)

Name: FLEX_SPI_TDR (FIFO_MULTI_DATA)
Offset: 0x40C
Reset: –
Property: Write-only

To write multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1 or 2), see SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TD1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TD1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:15, 16:31 – TDx Transmit Data
Next data to write in the Transmit FIFO. Information to be transmitted must be written to this register in a right-justified format.

63.10.44 SPI Status Register

Name: FLEX_SPI_SR
Offset: 0x410
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								SPIENS
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – RXFPTEF Receive FIFO Pointer Error Flag
See FIFO Pointer Error for details.
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. The receiver must be reset.

Bit 30 – TXFPTEF Transmit FIFO Pointer Error Flag
See FIFO Pointer Error for details.
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. The transceiver must be reset.

Bit 29 – RXFTHF Receive FIFO Threshold Flag
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold or RXFTH flag has been cleared.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold (changing states from "below threshold" to "equal to or above threshold").

Bit 28 – RXFFF Receive FIFO Full Flag
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been filled (changing states from "not full" to "full").

Bit 27 – RXFEF Receive FIFO Empty Flag
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been emptied (changing states from “not empty” to “empty”).

Bit 26 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_SPI_SR.

Bit 25 – TXFFF Transmit FIFO Full Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not full or TXFF flag has been cleared.
1	Transmit FIFO has been filled since the last read of FLEX_SPI_SR.

Bit 24 – TXFEF Transmit FIFO Empty Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_SPI_SR.

Bit 16 – SPIENS SPI Enable Status

Value	Description
0	SPI is disabled.
1	SPI is enabled.

Bit 12 – SFERR Client Mode Frame Error (cleared on read)

Value	Description
0	No frame error has been detected for a client access since the last read of FLEX_SPI_SR.
1	In Client mode, the chip select raised while the character defined in FLEX_SPI_CSR0.BITS was not complete.

Bit 11 – CMP Comparison Status (cleared on read)

Value	Description
0	No received character matched the comparison criteria programmed in VAL1 and VAL2 fields in FLEX_SPI_CMPR since the last read of FLEX_SPI_SR.
1	A received character matched the comparison criteria since the last read of FLEX_SPI_SR.

Bit 10 – UNDES Underrun Error Status (Client mode only) (cleared on read)

Value	Description
0	No underrun has been detected since the last read of FLEX_SPI_SR.
1	A transfer starts whereas no data has been loaded in FLEX_SPI_TDR, cleared when FLEX_SPI_SR is read.

Bit 9 – TXEMPTY Transmission Registers Empty (cleared by writing FLEX_SPI_TDR)

Value	Description
0	As soon as data is written in FLEX_SPI_TDR.
1	FLEX_SPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

Bit 8 – NSSR NSS Rising (cleared on read)

Value	Description
0	No rising edge detected on NSS pin since the last read of FLEX_SPI_SR.

Value	Description
1	A rising edge occurred on NSS pin since the last read of FLEX_SPI_SR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when FLEX_SPI_RDR is loaded at least twice from the shift register since the last read of FLEX_SPI_RDR.

Value	Description
0	No overrun has been detected since the last read of FLEX_SPI_SR.
1	An overrun has occurred since the last read of FLEX_SPI_SR.

Bit 2 – MODF Mode Fault Error (cleared on read)

Value	Description
0	No mode fault has been detected since the last read of FLEX_SPI_SR.
1	A mode fault occurred since the last read of FLEX_SPI_SR.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing FLEX_SPI_TDR)
When FIFOs are disabled:
0: Data has been written to FLEX_SPI_TDR and not yet transferred to the internal shift register.
1: The last data written to FLEX_SPI_TDR has been transferred to the internal shift register.
TDRE is cleared when the SPI is disabled or at reset. Enabling the SPI sets the TDRE flag.
When FIFOs are enabled:
0: Transmit FIFO cannot accept more data.
1: Transmit FIFO can accept data; one or more data can be written according to TXRDYM field configuration.
TDRE behavior with FIFOs enabled is illustrated in 63.8.7.5. TXEMPTY, TDRE and RDRF Behavior.

Bit 0 – RDRF Receive Data Register Full (cleared by reading FLEX_SPI_RDR)
When FIFOs are disabled:
0: No data has been received since the last read of FLEX_SPI_RDR.
1: Data has been received and the received data has been transferred from the internal shift register to FLEX_SPI_RDR since the last read of FLEX_SPI_RDR.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RDRF behavior with FIFOs enabled is illustrated in 63.8.7.5. TXEMPTY, TDRE and RDRF Behavior.

63.10.45 SPI Interrupt Enable Register

Name: FLEX_SPI_IER
Offset: 0x414
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				W	W	W	W	W
Reset				–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Enable

Bit 30 – TXFPTEF TXFPTEF Interrupt Enable

Bit 29 – RXFTHF RXFTHF Interrupt Enable

Bit 28 – RXFFF RXFFF Interrupt Enable

Bit 27 – RXFEF RXFEF Interrupt Enable

Bit 26 – TXFTHF TXFTHF Interrupt Enable

Bit 25 – TXFFF TXFFF Interrupt Enable

Bit 24 – TXFEF TXFEF Interrupt Enable

Bit 12 – SFERR Client Mode Frame Error Interrupt Enable

Bit 11 – CMP Comparison Interrupt Enable

Bit 10 – UNDES Underrun Error Interrupt Enable

Bit 9 – TXEMPTY Transmission Registers Empty Enable

Bit 8 – NSSR NSS Rising Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – MODF Mode Fault Error Interrupt Enable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

63.10.46 SPI Interrupt Disable Register

Name: FLEX_SPI_IDR
Offset: 0x418
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Disable

Bit 30 – TXFPTEF TXFPTEF Interrupt Disable

Bit 29 – RXFTHF RXFTHF Interrupt Disable

Bit 28 – RXFFF RXFFF Interrupt Disable

Bit 27 – RXFEF RXFEF Interrupt Disable

Bit 26 – TXFTHF TXFTHF Interrupt Disable

Bit 25 – TXFFF TXFFF Interrupt Disable

Bit 24 – TXFEF TXFEF Interrupt Disable

Bit 12 – SFERR Client Mode Frame Error Interrupt Disable

Bit 11 – CMP Comparison Interrupt Disable

Bit 10 – UNDES Underrun Error Interrupt Disable

Bit 9 – TXEMPTY Transmission Registers Empty Disable

Bit 8 – NSSR NSS Rising Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – MODF Mode Fault Error Interrupt Disable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

63.10.47 SPI Interrupt Mask Register

Name: FLEX_SPI_IMR
Offset: 0x41C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – RXFPTEF RXFPTEF Interrupt Mask

Bit 30 – TXFPTEF TXFPTEF Interrupt Mask

Bit 29 – RXFTHF RXFTHF Interrupt Mask

Bit 28 – RXFFF RXFFF Interrupt Mask

Bit 27 – RXFEF RXFEF Interrupt Mask

Bit 26 – TXFTHF TXFTHF Interrupt Mask

Bit 25 – TXFFF TXFFF Interrupt Mask

Bit 24 – TXFEF TXFEF Interrupt Mask

Bit 12 – SFERR Client Mode Frame Error Interrupt Mask

Bit 11 – CMP Comparison Interrupt Mask

Bit 10 – UNDES Underrun Error Interrupt Mask

Bit 9 – TXEMPTY Transmission Registers Empty Mask

Bit 8 – NSSR NSS Rising Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – MODF Mode Fault Error Interrupt Mask

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

63.10.48 SPI Chip Select Register

Name: FLEX_SPI_CSRx
Offset: 0x0430 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

FLEX_SPI_CSRx must be written even if the user wants to use the default reset values. The BITS field is not updated with the translated value unless the register is written.

Bit	31	30	29	28	27	26	25	24
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCT = Delay Between Consecutive Transfers × $f_{\text{peripheral clock}}$ / 32
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCT = Delay Between Consecutive Transfers × f_{GCLK} / 32

Bits 23:16 – DLYBS[7:0] Delay Before SPCK
This field defines the delay from NPCS falling edge (activation) to the first valid SPCK transition.
When DLYBS = 0, the delay is half the SPCK clock period.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBS = Delay Before SPCK × $f_{\text{peripheral clock}}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBS = Delay Before SPCK × f_{GCLK}

Bits 15:8 – SCBR[7:0] Serial Clock Bit Rate
In Host mode, the SPI Interface uses a modulus counter to derive the SPCK bit rate from the clock defined by the bit BRSRCCLK. The bit rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK bit rate:
If FLEX_SPI_MR.BRSRCCLK = 0: SCBR = $f_{\text{peripheral clock}}$ / SPCK Bit Rate
If FLEX_SPI_MR.BRSRCCLK = 1: SCBR = f_{GCLK} / SPCK Bit Rate
Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.
If BRSRCCLK = 1 in FLEX_SPI_MR, SCBR must be programmed with a value greater than 1.
At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.

Note: If one of the FLEX_SPI_CSRx.SCBR fields is set to 1, the other FLEX_SPI_CSRx.SCBR fields must be set to 1 as well, if they are used to process transfers. If they are not used to transfer data, they can be set at any value.

Bits 7:4 – BITS[3:0] Bits Per Transfer
See Note.
The BITS field determines the number of data bits transferred. Reserved values should not be used.

Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9-15	Reserved	

Bit 3 – CSAAT Chip Select Active After Transfer

Value	Description
0	The Peripheral Chip Select Line rises as soon as the last transfer is achieved.
1	The Peripheral Chip Select does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.

Bit 2 – CSNAAT Chip Select Not Active After Transfer (Ignored if CSAAT = 1)

If FLEX_SPI_MR.BRSRCCLK = 0: $\frac{\mathrm{DLYBCS}}{\mathrm{f}_{\mathrm{peripheral\ clock}}}$(if DLYBCS ≠ 0)

If FLEX_SPI_MR.BRSRCCLK = 1: $\frac{\mathrm{DLYBCS}}{\mathrm{f}_{GCLK}}$

If DLYBCS < 6, a minimum of six periods is introduced.

Value	Description
0	The Peripheral Chip Select does not rise between two transfers if the FLEX_SPI_TDR is reloaded before the end of the first transfer and if the two transfers occur on the same Chip Select.
1	The Peripheral Chip Select rises systematically after each transfer performed on the same client. It remains inactive after the end of transfer for a minimal duration of:

Bit 1 – NCPHA Clock Phase
NCPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data are changed on the leading edge of SPCK and captured on the following edge of SPCK.
1	Data are captured on the leading edge of SPCK and changed on the following edge of SPCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of SPCK is logic level zero.
1	The inactive state value of SPCK is logic level one.

63.10.49 SPI FIFO Mode Register

Name: FLEX_SPI_FMR
Offset: 0x440
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-32	Defines the Receive FIFO threshold value (number of data). The FLEX_SPI_SR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-32	Defines the Transmit FIFO threshold value (number of data). The FLEX_SPI_SR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receive Data Register Full Mode
If FIFOs are enabled, the FLEX_SPI_SR.RDRF flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RDRF will be at level '1' when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 byte must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 1 halfword must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte or halfword depending on FLEX_SPI_CSR0.BITS.

Value	Name	Description
1	TWO_DATA	RDRF will be at level ‘1’ when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 1 word (1 word carries 2 halfwords) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as halfword (2 bytes per access, 1 access when FLEX_SPI_CSR0.BITS=0), or word (2 halfwords per access, 2 accesses when FLEX_SPI_CSR0.BITS>0).
2	FOUR_DATA	RDRF will be at level ‘1’ when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 word (1 halfword carries 4 bytes) must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 2 words (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as word (4 bytes per access, 1 access when FLEX_SPI_CSR0.BITS=0 or 2 halfwords per access, 2 accesses when FLEX_SPI_CSR0.BITS>0).

Bits 1:0 – TXRDYM[1:0] Transmit Data Register Empty Mode
If FIFOs are enabled, the FLEX_SPI_SR.TDRE flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TDRE will be at level ‘1’ when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data, the chunk of 1 data (byte or halfword) must be configured in the DMA depending on FLEX_SPI_CSR0.BITS. If the transfer is performed by software, the access type (byte, halfword) must be defined depending on FLEX_SPI_CSR0.BITS.
1	TWO_DATA	TDRE will be at level ‘1’ when at least two data can be written in the transmit FIFO. FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 2 data) must be configured in the DMA. If the transfer is performed by software, the access type must be defined as word (2 data per access, 1 access).

63.10.50 SPI FIFO Level Register

Name: FLEX_SPI_FLR
Offset: 0x444
Reset: 0x00000000
Property: Read-only

This register reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–32	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–32	Indicates the number of data in the Transmit FIFO.

63.10.51 SPI Comparison Register

Name: FLEX_SPI_CMPR
Offset: 0x448
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VAL2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VAL1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – VAL2[15:0] Second Comparison Value for Received Character

Value	Description
0-65535	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_SPI_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the SPI requests a system wakeup if condition is met.

Bits 15:0 – VAL1[15:0] First Comparison Value for Received Character

Value	Description
0-65535	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_SPI_SR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the SPI requests a system wakeup if the condition is met.

63.10.52 SPI Write Protection Mode Register

Name: FLEX_SPI_WPMR
Offset: 0x4E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535049	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 0 – WPEN Write Protection Enable
See 63.8.8. SPI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

63.10.53 SPI Write Protection Status Register

Name: FLEX_SPI_WPSR
Offset: 0x4E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protect violation has occurred since the last read of FLEX_SPI_WPSR.
1	A write protect violation has occurred since the last read of FLEX_SPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

63.10.54 TWI Control Register

Name: FLEX_TWI_CR
Offset: 0x600
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TWI Write Protection Mode register.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		LOCKCLR		THRCLR
Access			W	W		W		W
Reset			–	–		–		–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs

Bit 26 – LOCKCLR Lock Clear

Value	Description
0	No effect.
1	Clear the TWI FSM lock.

Bit 24 – THRCLR Transmit Holding Register Clear

Value	Description
0	No effect.
1	Clear the Transmit Holding register and set TXRDY, TXCOMP flags.

Bit 17 – ACMDIS Alternative Command Mode Disable

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

Value	Description
0	No effect.
1	If Host mode is enabled, send a bus clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

Value	Description
0	No effect.
1	High-speed mode enabled.

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Value	Description
0	No effect.
1	If Host mode is enabled, an SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Client Mode Disabled

Value	Description
0	No effect.
1	Client mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Client Mode Enabled

Switching from Host to Client mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Client mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Host Mode Disabled

Value	Description
0	No effect.
1	Host mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in case of write operation. In read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Host Mode Enabled

Switching from Client to Host mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Host mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Host Read mode. – In single data byte host read, both START and STOP must be set. – In multiple data bytes host read, the STOP must be set after the last data received but one. – In Host Read mode, if a NACK bit is received, the STOP is automatically performed. – In host data write operation, a STOP condition will be sent after the transmission of the current data is finished.

Bit 0 – START Send a START Condition

This action is necessary when the TWI peripheral needs to read data from a client. When configured in Host mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding register (FLEX_TWI_THR).

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Host Mode register (FLEX_TWI_MMR).

63.10.55 TWI Control Register (FIFO_ENABLED)

Name: FLEX_TWI_CR (FIFO_ENABLED)
Offset: 0x600
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), see 63.9.6.8. TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
Access			W	W		W	W	W
Reset			–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs.

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 17 – ACMDIS Alternative Command Mode Disable

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

Value	Description
0	No effect.
1	If Host mode is enabled, send a bus clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

Value	Description
0	No effect.
1	High-speed mode enabled.

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Value	Description
0	No effect.
1	If Host mode is enabled, a SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Client Mode Disabled

Value	Description
0	No effect.
1	Client mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Client Mode Enabled
Switching from Host to Client mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Client mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Host Mode Disabled

Value	Description
0	No effect.
1	Host mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in the case of a write operation. In a read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Host Mode Enabled
Switching from Client to Host mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Host mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Host Read mode. – In single data byte host read, both START and STOP must be set. – In multiple data bytes host read, the STOP must be set after the last data received but one. – In Host Read mode, if a NACK bit is received, the STOP is automatically performed. – In host data write operation, a STOP condition will be sent after the transmission of the current data is finished.

Bit 0 – START Send a START Condition
This action is necessary when the TWI peripheral needs to read data from a client. When configured in Host mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding register (FLEX_TWI_THR).

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Host Mode register (FLEX_TWI_MMR).

63.10.56 TWI Host Mode Register

Name: FLEX_TWI_MMR
Offset: 0x604
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								NOAP
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
		DADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SCLRBL[1:0]		MREAD			IADRSZ[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 24 – NOAP No Auto-Stop On NACK Error

Value	Description
0	A stop condition is sent automatically upon Not-Acknowledge error detection.
1	No automatic action is performed upon Not-Acknowledge error detection.

Bits 22:16 – DADR[6:0] Device Address
The device address is used to access client devices in Read or Write mode. Those bits are only used in Host mode.

Bits 14:13 – SCLRBL[1:0] SCL Rise Boost Level
Number of clock periods during which SCL rise is boosted (meaning line driven to level ‘1’).

Bit 12 – MREAD Host Read Direction

Value	Description
0	Host write direction.
1	Host read direction.

Bits 9:8 – IADRSZ[1:0] Internal Device Address Size

Value	Name	Description
0	NONE	No internal device address
1	1_BYTE	One-byte internal device address
2	2_BYTE	Two-byte internal device address
3	3_BYTE	Three-byte internal device address

63.10.57 TWI Client Mode Register

Name: FLEX_TWI_SMR
Offset: 0x608
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DATAMEN	SADR3EN	SADR2EN	SADR1EN				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
		SADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
		MASK[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 31 – DATAMEN Data Matching Enable

Value	Description
0	Data matching on first received data is disabled.
1	Data matching on first received data is enabled.

Bit 30 – SADR3EN Client Address 3 Enable

Value	Description
0	Client address 3 matching is disabled.
1	Client address 3 matching is enabled.

Bit 29 – SADR2EN Client Address 2 Enable

Value	Description
0	Client address 2 matching is disabled.
1	Client address 2 matching is enabled.

Bit 28 – SADR1EN Client Address 1 Enable

Value	Description
0	Client address 1 matching is disabled.
1	Client address 1 matching is enabled.

Bits 22:16 – SADR[6:0] Client Address
The client device address is used in Client mode in order to be accessed by host devices in Read or Write mode. SADR must be programmed before enabling Client mode or after a general call. Writes at other times have no effect.

Bits 14:8 – MASK[6:0] Client Address Mask
A mask can be applied on the client device address in Client mode in order to allow multiple address answer. For each bit of the MASK field set to one, the corresponding SADR bit will be masked.
If the MASK field is set to 0, no mask is applied to the SADR field.

Bit 7 – SNIFF Client Sniffer Mode

Value	Description
0	Client Sniffer mode is disabled.
1	Client Sniffer mode is enabled.

Bit 6 – SCLWSDIS Clock Wait State Disable

Value	Description
0	No effect.
1	Clock stretching disabled in Client mode, OVRE and UNRE will indicate overrun and underrun.

Bit 5 – BSEL TWI Bus Selection

Value	Description
0	TWI analyzes the TWCK and TWD pins from its TWI bus.
1	TWI analyzes the TWCK pins TWD from consecutive index TWI peripheral of the product.

Bit 4 – SADAT Client Address Treated as Data
When Client Sniffer Mode is enabled, the client address is always received as data in FLEX_TWI_RHR and SADAT has no effect.

Value	Description
0	Client address is handled normally (will not trig RXRDY flag and will not fill FLEX_TWI_RHR upon reception).
1	Client address is handled as data field, RXRDY will be set and FLEX_TWI_RHR filled upon client address reception.

Bit 3 – SMHH SMBus Host Header

Value	Description
0	Acknowledge of the SMBus Host Header disabled.
1	Acknowledge of the SMBus Host Header enabled.

Bit 2 – SMDA SMBus Default Address

Value	Description
0	Acknowledge of the SMBus Default Address disabled.
1	Acknowledge of the SMBus Default Address enabled.

Bit 0 – NACKEN Client Receiver Data Phase NACK Enable

Value	Description
0	Normal value to be returned in the ACK cycle of the data phase in Client Receiver mode.
1	NACK value to be returned in the ACK cycle of the data phase in Client Receiver mode.

63.10.58 TWI Internal Address Register

Name: FLEX_TWI_IADR
Offset: 0x60C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	IADR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	IADR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	IADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – IADR[23:0] Internal Address
0, 1, 2 or 3 bytes depending on IADRSZ.

63.10.59 TWI Clock Waveform Generator Register

Name: FLEX_TWI_CWGR
Offset: 0x610
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

FLEX_TWI_CWGR is only used in Host mode.

Bit	31	30	29	28	27	26	25	24
			HOLD[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				BRSRCCLK		CKDIV[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	15	14	13	12	11	10	9	8
	CHDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – HOLD[5:0] TWD Hold Time Versus TWCK Falling
If High-speed mode is selected TWD is internally modified on the TWCK falling edge to meet the I2C specified maximum hold time, else if High-speed mode is not configured TWD is kept unchanged after TWCK falling edge for a period of (HOLD + 3) × $t_{peripheral\ clock}$.

Bit 20 – BRSRCCLK Bit Rate Source Clock

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bits 18:16 – CKDIV[2:0] Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – CHDIV[7:0] Clock High Divider
The SCL high period is defined as follows:
If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $(t_{high}/t_{ext\_ck})/2^{CKDIV}$

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3 - (THRES+1))/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $((t_{high} - (THRES+1) * t_{peripheral\ clock})/t_{ext\_ck})/2^{CKDIV}$

Bits 7:0 – CLDIV[7:0] Clock Low Divider
The SCL low period is defined as follows:

If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CLDIV = $((t_{low}/t_{peripheral\ clock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CLDIV = $(t_{low}/t_{ext\_ck})/2^{CKDIV}$

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CLDIV = $((t_{low}/t_{peripheral\ clock}) - 3 - (THRES+1))/2^{CKDIV}$
- If BRSRCCLK = 1: CLDIV = $((t_{low} - (THRES+1) * t_{peripheral\ clock})/t_{ext\_ck})/2^{CKDIV}$

63.10.60 TWI Status Register

Name: FLEX_TWI_SR
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	LOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					EOSACC	SCLWS	ARBLST	NACK
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – LOCK TWI Lock Due to Frame Errors

Value	Description
0	The TWI is not locked.
1	The TWI is locked due to frame errors (see 63.9.3.13. Handling Errors in Alternative Command and 63.9.6. TWI FIFOs).

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 17 – SMBAF SMBus Alert Flag (cleared on read)

Value	Description
0	No SMBus client drives the SMBALERT line.
1	At least one SMBus client drives the SMBALERT line.

Bit 16 – MCACK Host Code Acknowledge (cleared on read)
MACK used in Client mode:

Value	Description
0	No host code has been received.
1	A host code has been received.

Bit 11 – EOSACC End Of Client Access (cleared on read)
This bit is only used in Client mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A client access is being performed.
1	Client Access is finished. End Of Client Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State
This bit is only used in Client mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)
This bit is only used in Host mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another host of the TWI bus has won the multi-host arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)
NACK used in Host mode:
0: Each data byte has been correctly received by the far-end side TWI client component.
1: A data or address byte has not been acknowledged by the client component. Set at the same time as TXCOMP.
NACK used in Client Read mode:
0: Each data byte has been correctly received by the host.
1: In Read mode, a data byte has not been acknowledged by the host. When NACK is set, the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the host will stop the data transfer or reinitiate it.
Note that in Client Write mode, all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)
This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)
This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

Bit 5 – GACC General Call Access (cleared on read)
This bit is only used in Client mode.
GACC behavior can be seen in figure Host Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Client Access
This bit is only used in Client mode.
SVACC behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a host has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Client Read
This bit is only used in Client mode. When SVACC is low (no client access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a host.
1	Indicates that a read access is performed by a host.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)
TXRDY used in Host mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Host mode can be seen in figures Host Write with One Data Byte,Host Write with Multiple Data Bytes and Host Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Client mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Client mode can be seen in figures Read Access Ordered by a Host, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)
When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Host mode can be seen in figure Host Read with Multiple Data Bytes.
RXRDY behavior in Client mode can be seen in figures Write Access Ordered by a Host, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)
TXCOMP used in Host mode:
0: During the length of the current frame.
1: When both the holding register and the internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Host mode can be seen in figures Host Write with One Byte Internal Address and Multiple Data Bytes and Host Read with Multiple Data Bytes.
TXCOMP used in Client mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Client mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

63.10.61 TWI Status Register (FIFO ENABLED)

Name: FLEX_TWI_SR (FIFO_ENABLED)
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					EOSACC	SCLWS	ARBLST	NACK
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 17 – SMBAF SMBus Alert Flag (cleared on read)

Value	Description
0	No SMBus client drives the SMBALERT line.
1	At least one SMBus client drives the SMBALERT line.

Bit 16 – MCACK Host Code Acknowledge (cleared on read)
MACK used in Client mode:

Value	Description
0	No host code has been received.
1	A host code has been received.

Bit 11 – EOSACC End Of Client Access (cleared on read)
This bit is only used in Client mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A Client Access is being performed.
1	Client Access is finished. End Of Client Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State
This bit is only used in Client mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)
This bit is only used in Host mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another host of the TWI bus has won the multi-host arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)
NACK used in Host mode:
0: Each data byte has been correctly received by the far-end side TWI client component.
1: A data or address byte has not been acknowledged by the client component. Set at the same time as TXCOMP.
NACK used in Client Read mode:
0: Each data byte has been correctly received by the host.
1: In Read mode, a data byte has not been acknowledged by the host. When NACK is set the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the host will stop the data transfer or re initiate it.
Note that in Client Write mode all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)
This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)
This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

Bit 5 – GACC General Call Access (cleared on read)
This bit is only used in Client mode.
GACC behavior can be seen in figure Host Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Client Access
This bit is only used in Client mode.
SVACC behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a host has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Client Read
This bit is only used in Client mode. When SVACC is low (no client access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a host.
1	Indicates that a read access is performed by a host.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)
TXRDY used in Host mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Host mode can be seen in figures Host Write with One Data Byte,Host Write with Multiple Data Bytes and Host Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Client mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Client mode can be seen in figures Read Access Ordered by a Host, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)
When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Host mode can be seen in figure Host Read with Multiple Data Bytes.
RXRDY behavior in Client mode can be seen in figures Write Access Ordered by a Host, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)
TXCOMP used in Host mode:
0: During the length of the current frame.
1: When both holding register and internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Host mode can be seen in figures Host Write with One Byte Internal Address and Multiple Data Bytes and Host Read with Multiple Data Bytes.
TXCOMP used in Client mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Client mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

63.10.62 TWI Interrupt Enable Register

Name:	FLEX_TWI_IER
Offset:	0x624
Reset:	–
Property:	Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Enable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Enable

Bit 19 – PECERR PEC Error Interrupt Enable

Bit 18 – TOUT Timeout Error Interrupt Enable

Bit 16 – MCACK Host Code Acknowledge Interrupt Enable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 12 – ENDRX End of Receive Buffer Interrupt Enable

Bit 11 – EOSACC End Of Client Access Interrupt Enable

Bit 10 – SCL_WS Clock Wait State Interrupt Enable

Bit 9 – ARBLST Arbitration Lost Interrupt Enable

Bit 8 – NACK Not Acknowledge Interrupt Enable

Bit 7 – UNRE Underrun Error Interrupt Enable

Bit 6 – OVRE Overrun Error Interrupt Enable

Bit 5 – GACC General Call Access Interrupt Enable

Bit 4 – SVACC Client Access Interrupt Enable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Enable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Enable

Bit 0 – TXCOMP Transmission Completed Interrupt Enable

63.10.63 TWI Interrupt Disable Register

Name: FLEX_TWI_IDR
Offset: 0x628
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Disable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Disable

Bit 19 – PECERR PEC Error Interrupt Disable

Bit 18 – TOUT Timeout Error Interrupt Disable

Bit 16 – MCACK Host Code Acknowledge Interrupt Disable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 12 – ENDRX End of Receive Buffer Interrupt Disable

Bit 11 – EOSACC End Of Client Access Interrupt Disable

Bit 10 – SCL_WS Clock Wait State Interrupt Disable

Bit 9 – ARBLST Arbitration Lost Interrupt Disable

Bit 8 – NACK Not Acknowledge Interrupt Disable

Bit 7 – UNRE Underrun Error Interrupt Disable

Bit 6 – OVRE Overrun Error Interrupt Disable

Bit 5 – GACC General Call Access Interrupt Disable

Bit 4 – SVACC Client Access Interrupt Disable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Disable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Disable

Bit 0 – TXCOMP Transmission Completed Interrupt Disable

63.10.64 TWI Interrupt Mask Register

Name: FLEX_TWI_IMR
Offset: 0x62C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			R	R	R	R		R
Reset			0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Mask

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Mask

Bit 19 – PECERR PEC Error Interrupt Mask

Bit 18 – TOUT Timeout Error Interrupt Mask

Bit 16 – MCACK Host Code Acknowledge Interrupt Mask

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 14 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 13 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 12 – ENDRX End of Receive Buffer Interrupt Mask

Bit 11 – EOSACC End Of Client Access Interrupt Mask

Bit 10 – SCL_WS Clock Wait State Interrupt Mask

Bit 9 – ARBLST Arbitration Lost Interrupt Mask

Bit 8 – NACK Not Acknowledge Interrupt Mask

Bit 7 – UNRE Underrun Error Interrupt Mask

Bit 6 – OVRE Overrun Error Interrupt Mask

Bit 5 – GACC General Call Access Interrupt Mask

Bit 4 – SVACC Client Access Interrupt Mask

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Mask

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Mask

Bit 0 – TXCOMP Transmission Completed Interrupt Mask

63.10.65 TWI Receive Holding Register

Name: FLEX_TWI_RHR
Offset: 0x630
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN=1), a byte access on FLEX_TWI_RHR reads one data, see TWI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 12:11 – ASTATE[1:0] Acknowledge State (Client Sniffer Mode only)

Value	Name	Description
0	NONE	No Acknowledge or Nacknowledge detected after previously logged data
1	ACK	Acknowledge (A) detected after previously logged data
2	NACK	Nacknowledge (NA) detected after previously logged data
3	UNDEF	Not defined

Bit 10 – PSTATE Stop State (Client Sniffer Mode only)

Value	Description
0	No STOP (P) detected after previous logged data.
1	Stop detected (P) after previous logged data.

Bits 9:8 – SSTATE[1:0] Start State (Client Sniffer Mode only)

Value	Name	Description
0	NOSTART	No START detected with the logged data
1	START	START (S) detected with the logged data
2	RSTART	Repeated START (Sr) detected with the logged data
3	UNDEF	Not defined

Bits 7:0 – RXDATA[7:0] Host or Client Receive Holding Data

63.10.66 TWI Receive Holding Register (FIFO Enabled)

Name: FLEX_TWI_RHR (FIFO_ENABLED)
Offset: 0x630
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode disabled (FLEX_TWI_SMR.SNIFF=0). The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RXDATA3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RXDATA2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXDATA1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RXDATA0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – RXDATA3[7:0] Host or Client Receive Holding Data 3

Bits 23:16 – RXDATA2[7:0] Host or Client Receive Holding Data 2

Bits 15:8 – RXDATA1[7:0] Host or Client Receive Holding Data 1

Bits 7:0 – RXDATA0[7:0] Host or Client Receive Holding Data 0

63.10.67 TWI Transmit Holding Register

Name: FLEX_TWI_THR
Offset: 0x634
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN=1), a byte access on FLEX_TWI_THR reads one data in a single access, see TWI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXDATA[7:0] Host or Client Transmit Holding Data

63.10.68 TWI Transmit Holding Register (FIFO Enabled)

Name: FLEX_TWI_THR (FIFO_ENABLED)
Offset: 0x634
Reset: –
Property: Write-only

To write multi-data, the FIFO must be enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode disabled (FLEX_TWI_SMR.SNIFF=0). The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TXDATA3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TXDATA2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXDATA1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDATA0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – TXDATA3[7:0] Host or Client Transmit Holding Data 3

Bits 23:16 – TXDATA2[7:0] Host or Client Transmit Holding Data 2

Bits 15:8 – TXDATA1[7:0] Host or Client Transmit Holding Data 1

Bits 7:0 – TXDATA0[7:0] Host or Client Transmit Holding Data 0

63.10.69 TWI SMBus Timing Register

Name: FLEX_TWI_SMBTR
Offset: 0x638
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	THMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TLOWM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TLOWS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					PRESC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:24 – THMAX[7:0] Clock High Maximum Cycles
Clock cycles in clock high maximum count. Prescaled by PRESC. Used for bus free detection. Used to time THIGH:MAX.

Bits 23:16 – TLOWM[7:0] Main System Bus Clock Stretch Maximum Cycles

Value	Description
0	TLOW:MEXT timeout check disabled.
1-255	Clock cycles in main system bus maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:MEXT.

Bits 15:8 – TLOWS[7:0] Client Clock Stretch Maximum Cycles

Value	Description
0	TLOW:SEXT timeout check disabled.
1-255	Clock cycles in client maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:SEXT.

Bits 3:0 – PRESC[3:0] SMBus Clock Prescaler
Used to specify how to prescale the TLOWS, TLOWM and THMAX counters in SMBTR. Counters are prescaled according to the following formula: PRESC = Log(fMCK / $f_{Prescaled}$) / Log(2) - 1

63.10.70 TWI High-Speed Register

Name: FLEX_TWI_HSR
Offset: 0x63C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	MCODE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – MCODE[7:0] TWI High-Speed Host Code

63.10.71 TWI Alternative Command Register

Name: FLEX_TWI_ACR
Offset: 0x640
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							NPEC	NDIR
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	NDATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							PEC	DIR
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	DATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – NPEC Next PEC Request (SMBus Mode only)

Value	Description
0	The next transfer does not use a PEC byte.
1	The next transfer uses a PEC byte.

Bit 24 – NDIR Next Transfer Direction

Value	Description
0	Write direction.
1	Read direction.

Bits 23:16 – NDATAL[7:0] Next Data Length

Value	Description
0	No data to send (see 63.9.3.12. Alternative Command).
1-255	Number of bytes to send for the next transfer.

Bit 9 – PEC PEC Request (SMBus Mode only)

Value	Description
0	The transfer does not use a PEC byte.
1	The transfer uses a PEC byte.

Bit 8 – DIR Transfer Direction

Value	Description
0	Write direction.
1	Read direction.

Bits 7:0 – DATAL[7:0] Data Length

Value	Description
0	No data to send (see 63.9.3.12. Alternative Command).

Value	Description
1–255	Number of bytes to send during the transfer.

63.10.72 TWI Filter Register

Name: FLEX_TWI_FILTR
Offset: 0x644
Reset: 0x00000000
Property: Read/Write

Important:
FILT and THRES are used to configure digital filters on data and clock lines.

In Standard, Fast and Fast Plus modes, the digital filter must be enabled (FILT=1) and a pulse width threshold defined (THRES > 0).

The field THRES must be set according to the peripheral clock to suppress spikes lower than 50 ns. The recommended value is calculated using the formula below:

THRES > 50 ns/$t_{peripheral\_clock\ (ns)}$

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						THRES[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
							PADFEN	FILT
Access							R/W	R/W
Reset							0	0

Bits 10:8 – THRES[2:0] Digital Filter Threshold

Value	Description
0	No filtering applied on TWI inputs.
1-7	Maximum pulse width of spikes which will be suppressed by the input filter, defined in peripheral clock cycles.

Bit 1 – PADFEN PAD Filter Enable

Value	Description
0	PAD analog filter is disabled.
1	PAD analog filter is enabled. (The analog filter must be enabled if High-speed mode is enabled.)

Bit 0 – FILT RX Digital Filter
TWI digital input filtering follows a majority decision based on three samples from SDA/SCL lines at peripheral clock frequency.

Value	Description
0	No filtering applied on TWI inputs.
1	TWI input filtering is active. (Only in Standard and Fast modes)

63.10.73 TWI High-Speed Clock Waveform Generator Register

Name: FLEX_TWI_HSCWGR
Offset: 0x648
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

FLEX_TWI_HSCWGR is only used in High-Speed Host mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
						HSCKDIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	15	14	13	12	11	10	9	8
	HSCHDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	HSCLDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – HSCKDIV[2:0] High-Speed Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – HSCHDIV[7:0] High-Speed Clock High Divider
The SCL high period is defined as follows:

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheralclock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $(t_{high}/t_{ext\_ck})/2^{CKDIV}$

Bits 7:0 – HSCLDIV[7:0] High-Speed Clock Low Divider
The SCL low period is defined as follows:

- If BRSRCCLK = 0: CLDIV = $((t_{low}/t_{peripheralclock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CLDIV = $(t_{low}/t_{ext\_ck})/2^{CKDIV}$

63.10.74 TWI Matching Register

Name: FLEX_TWI_SWMR
Offset: 0x64C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DATAM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
		SADR3[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
		SADR2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		SADR1[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:24 – DATAM[7:0] Data Match
The TWI module will extend the matching process to the first received data, comparing it with DATAM if the DATAMEN bit is enabled.

Bits 22:16 – SADR3[6:0] Client Address 3
Client address 3. The TWI module will match on this additional address if the SADR3EN bit is enabled.

Bits 14:8 – SADR2[6:0] Client Address 2
Client address 2. The TWI module will match on this additional address if the SADR2EN bit is enabled.

Bits 6:0 – SADR1[6:0] Client Address 1
Client address 1. The TWI module will match on this additional address if the SADR1EN bit is enabled.

63.10.75 TWI FIFO Mode Register

Name: FLEX_TWI_FMR
Offset: 0x650
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).
This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-32	Defines the Receive FIFO threshold value (number of bytes). The FLEX_TWI_FSR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-32	Defines the Transmit FIFO threshold value (number of bytes). The FLEX_TWI_FSR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.RXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RXRDY will be at level '1' when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type (byte, halfword) must be defined accordingly.
1	TWO_DATA	RXRDY will be at level '1' when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).

Value	Name	Description
2	FOUR_DATA	RXRDY will be at level ‘1’ when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level ‘1’ when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data, the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as byte.
1	TWO_DATA	TXRDY will be at level ‘1’ when at least two data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	TXRDY will be at level ‘1’ when at least four data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

63.10.76 TWI FIFO Level Register

Name: FLEX_TWI_FLR
Offset: 0x654
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–32	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–32	Indicates the number of data in the Transmit FIFO.

63.10.77 TWI FIFO Status Register

Name: FLEX_TWI_FSR
Offset: 0x660
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See 63.9.6.10. FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See 63.9.6.10. FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 4 – RXFFF Receive FIFO Full Flag

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 3 – RXFEF Receive FIFO Empty Flag

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last read of FLEX_TWI_FSR.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared on read)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared on read)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_TWI_FSR.

63.10.78 TWI FIFO Interrupt Enable Register

Name: FLEX_TWI_FIER
Offset: 0x664
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

63.10.79 TWI FIFO Interrupt Disable Register

Name: FLEX_TWI_FIDR
Offset: 0x668
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

63.10.80 TWI FIFO Interrupt Mask Register

Name: FLEX_TWI_FIMR
Offset: 0x66C
Reset: 0x00000000
Property: Read-only

This register reads ‘0’ if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

63.10.81 TWI Write Protection Mode Register

Name: FLEX_TWI_WPMR
Offset: 0x6E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x545749	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 0 – WPEN Write Protection Enable
See TWI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

63.10.82 TWI Write Protection Status Register

Name: FLEX_TWI_WPSR
Offset: 0x6E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protect Violation Status

Value	Description
0	No Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR.
1	A Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

64. Quad Serial Peripheral Interface (QSPI)

64.1 Description

The Quad Serial Peripheral Interface (QSPI) is a synchronous serial data link that provides communication with external devices in Host mode.

The QSPI can be used in SPI mode to interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors, or in Serial Memory mode to interface to serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to RAM. The serial Flash memory mapping is seen in the system as other memories such as ROM, SRAM, DRAM, embedded Flash memory, etc.

With the support of the Quad SPI protocol, the QSPI allows the system to use high-performance serial Flash memories which are small and inexpensive, in place of larger and more expensive parallel Flash memories.

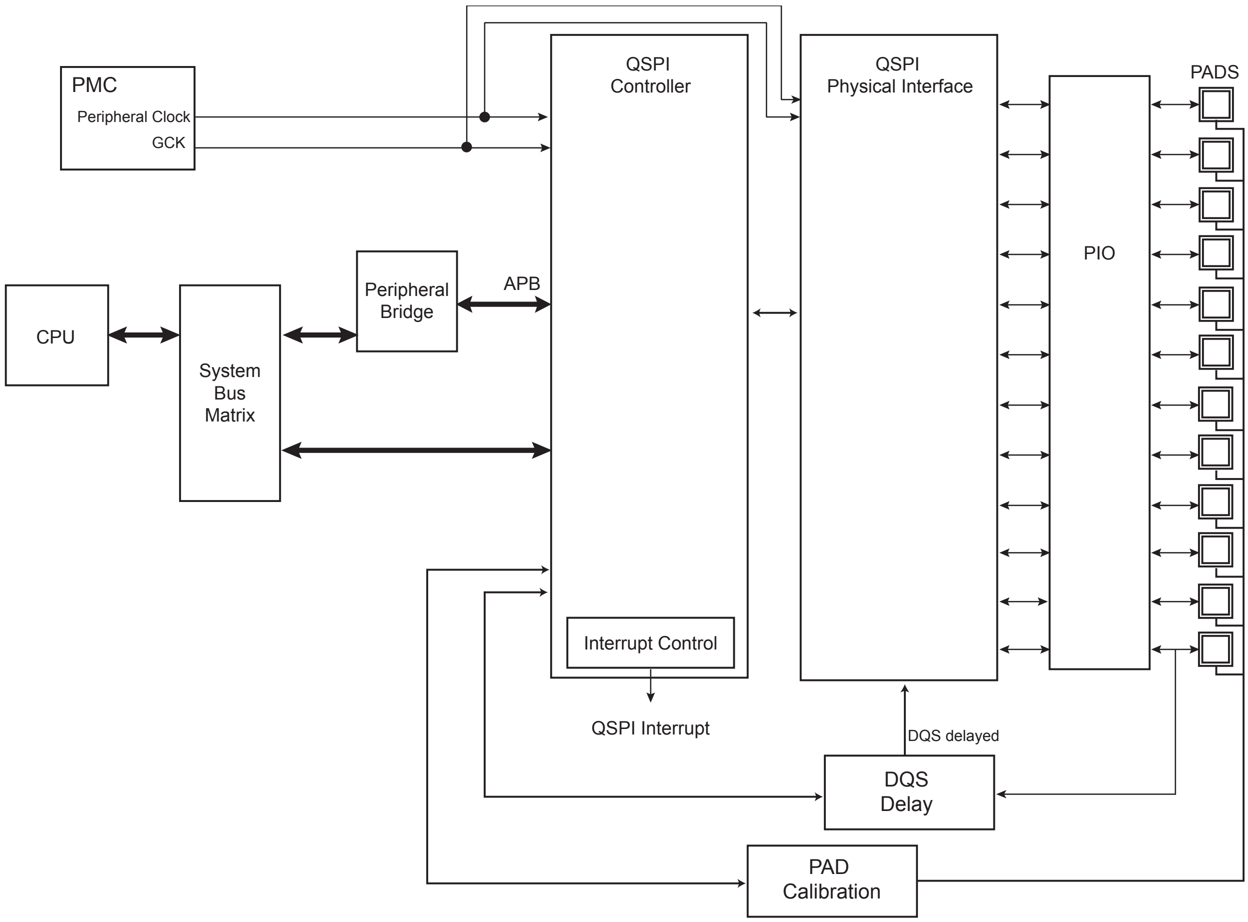
Note: Stacked devices with a rollover in the memory address space at each die boundary are not supported.

64.2 Embedded Characteristics

- Host SPI Interface
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode
  - Interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit/16-bit programmable data length
- Serial Memory Mode
  - Interface to serial Flash memories operating in Single-bit SPI, Dual SPI, Quad SPI and Octal SPI
  - Interface to serial Flash Memories operating in Single Data Rate or Double Data Rate Modes
  - Supports “Execute In Place” (XIP)— code execution by the system directly from a serial Flash memory
  - Flexible instruction register for compatibility with all serial Flash memories
  - 32-bit address mode (default is 8-bit address) to support serial Flash memories larger than 128 Mbits
  - Continuous Read mode
  - “On-The-Fly” Scrambling/unscrambling
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver, one channel for the transmitter
- Register Write Protection
- One Octal Serial Peripheral Interface (QSPI0) Supporting Up to 200 MHz DDR. Octal, TwinQuad, HyperFlash and OctaFlash Protocols Supported
- One Quad Serial Peripheral Interface (QSPI1) Supporting Up to 90 MHz DDR/133 MHz SDR

64.3 Block Diagram

Figure 64-1. QSPI Block Diagram



64.4 Signal Description

Table 64-1. Signal Description

Pin Name	Pin Description	Type
QSCK	Serial Clock	Output
QSPI0_SCKN	QSPI0 Negated Serial Clock for HyperFlash only	Output
MOSI (QIO0) (1) (2)	Data Output (Data Input Output 0)	Output (Input/Output)
MISO (QIO1) (1) (2)	Data Input (Data Input Output 1)	Input (Input/Output)
QIO2 (3)	Data Input Output 2	Input/Output
QIO3 (3)	Data Input Output 3	Input/Output
QIO4 (4)	Data Input Output 4	Input/Output
QIO5 (3)	Data Input Output 5	Input/Output
QIO6 (4)	Data Input Output 6	Input/Output
QIO7 (4)	Data Input Output 7	Input/Output

...........continued

Pin Name	Pin Description	Type
QCS	Peripheral Chip Select	Output
QINT (5)	Interrupt Input	Input
DQS (6)	Data Strobe	Input

Notes:
1. MOSI and MISO are used for single-bit SPI operation.
2. QIO0–QIO1 are used for Dual SPI operation.
3. QIO0–QIO3 are used for Quad SPI operation.
4. QIO4–QIO7 are used for Octal SPI operation.
5. QINT is supplied by some Octal SPI memories
6. DQS is supplied by most Octal SPI memories

64.5 Product Dependencies

64.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the QSPI pins to their peripheral functions.

64.5.2 Power Management

The QSPI may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the QSPI clocks.

64.5.3 Interrupt Sources

The QSPI has an interrupt line connected to the Interrupt Controller. Handling the QSPI interrupt requires programming the interrupt controller before configuring the QSPI.

64.5.4 Direct Memory Access Controller (DMA)

The QSPI can be used in conjunction with the Direct Memory Access Controller (DMA) in order to reduce processor overhead. For a full description of the DMA, refer to the section *DMA Controller (XDMAC)*.

64.6 Functional Description

64.6.1 Register Synchronization

Because the QSPI Controller interface and the QSPI Controller core use different clocks, some events must be synchronized with the core for them to take effect.

The events that require synchronization with the QSPI Controller core are:

- QSPI_CR.QSPIEN
- QSPI_CR.QSPIDIS
- QSPI_CR.SRFRSH
- QSPI_CR.SWRST
- QSPI_CR.UPDCFG
- QSPI_CR.STTFR
- QSPI_CR.RTOUT
- QSPI_CR.LASTXFER
- QSPI_RDR.RD (implies synchronization only when QSPI_MR.SMM is set to ‘1 ‘)

- QSPI_TDR.TD

Before accessing any of these bits/fields, check that QSPI_SR.SYNCBSY is at ‘0’.

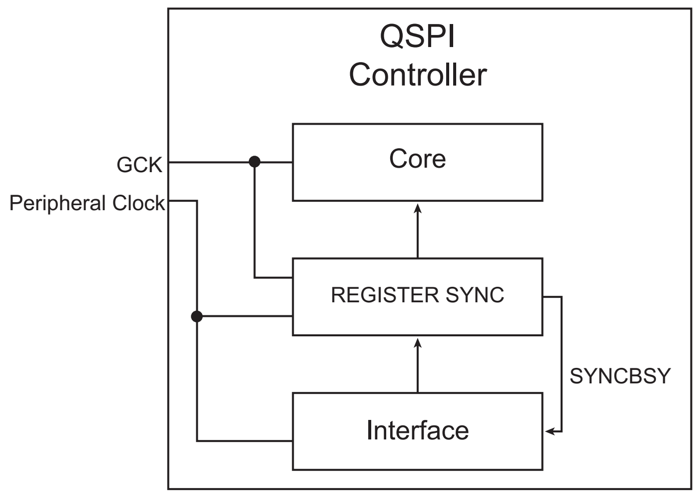
When the synchronization process is in progress, QSPI_SR.SYNCBSY is at ‘1’.

As long as QSPI_SR.SYNCBSY is at ‘1’, no access to the registers requiring synchronization is allowed.

When using QSPI_CR.UPDCFG to update the system configuration, it is mandatory for QSPI_SR.SYNCBSY to be at ‘0’ before and after writing QSPI_CR.UPDCFG. This ensures that the update is completed before going on to the next step.

Note: In some specific cases, it is not necessary to check QSPI_SR.SYNCBSY for QSPI_RDR.RD and QSPI_TDR.TD if the TDRE or RDRF flags are checked. See Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access), Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) and HyperFlash “Write Buffer” Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) for detailed procedures.

Figure 64-2. Register Synchronization with QSPI Controller Core



64.6.2 Updating the QSPI Configuration

At any time, the QSPI Controller core configuration can be updated by writing the QSPI_CR.UPDCFG bit to ‘1’. Note that QSPI_SR.SYNCBSY must be ‘0’ before writing QSPI_CR.UPDCFG.

The configuration registers that require synchronization (writing QSPI_CR.UPDCFG to ‘1’) with the QSPI Controller core are:

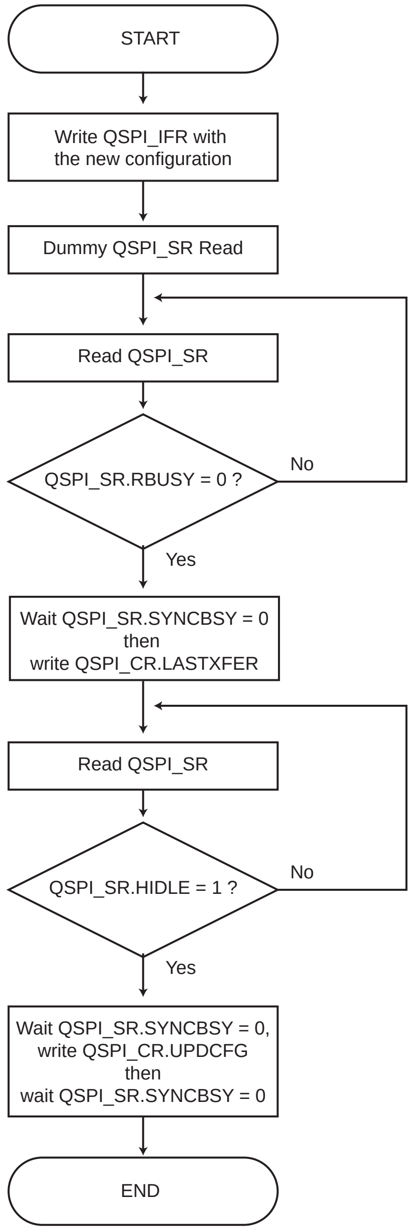
- QSPI_MR
- QSPI_SCR
- QSPI_IAR
- QSPI_WICR
- QSPI_IFR
- QSPI_RICR
- QSPI_SMR
- QSPI_SKR
- QSPI_REFRESH
- QSPI_WRACNT
- QSPI_PCALCFG

64.6.2.1 Changing QSPI_IFR Configuration (QSPI_MR.SMM = 1)

Accesses in Serial Memory mode (QSPI_MR.SMM=1) are triggered by QSPI register accesses or by performing accesses in the QSPI memory space depending on the configuration of QSPI_IFR.SMRM and QSPI_IFR.TFRTYP. For details, see Instruction Frame and QSPI Access Triggering.

If QSPI_IFR is to be modified after the initial configuration, the procedure illustrated in the figure below must be followed to ensure no frame loss. For details, see QSPI Access Triggering.

Figure 64-3. Changing QSPI_IFR



64.6.3 Serial Clock Phase and Polarity

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the QSPI Serial Clock register (QSPI_SCR). The CPHA bit in the QSPI_SCR programs the clock phase. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Thus, the interfaced client must use the same parameter values to communicate.

Only Mode 0 is supported with QSPI_MR.SMM = 1.

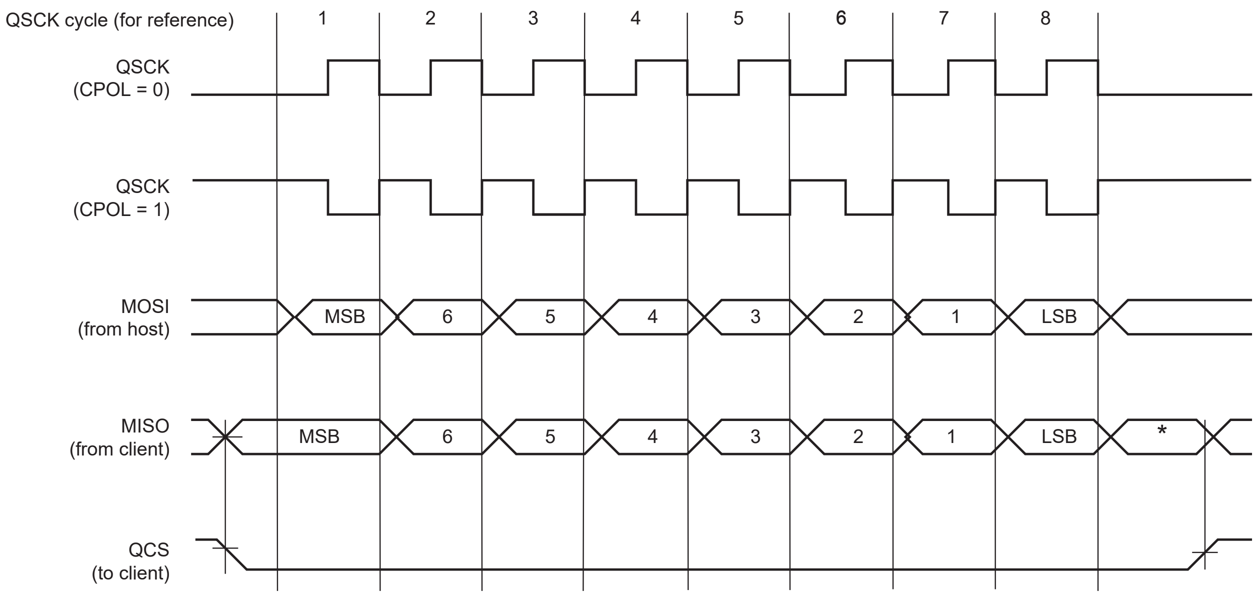
The table below shows the four modes and corresponding parameter settings.

Table 64-2. QSPI Bus Clock Modes

QSPI Clock Mode	QSPI_SCR.CPOL	QSPI_SCR.CPHA	Shift QSCK Edge	Capture QSCK Edge	QSCK Inactive Level
0	0	0	Falling	Falling	Low
1	0	1	Rising	Rising	Low
2	1	0	Rising	Rising	High
3	1	1	Falling	Falling	High

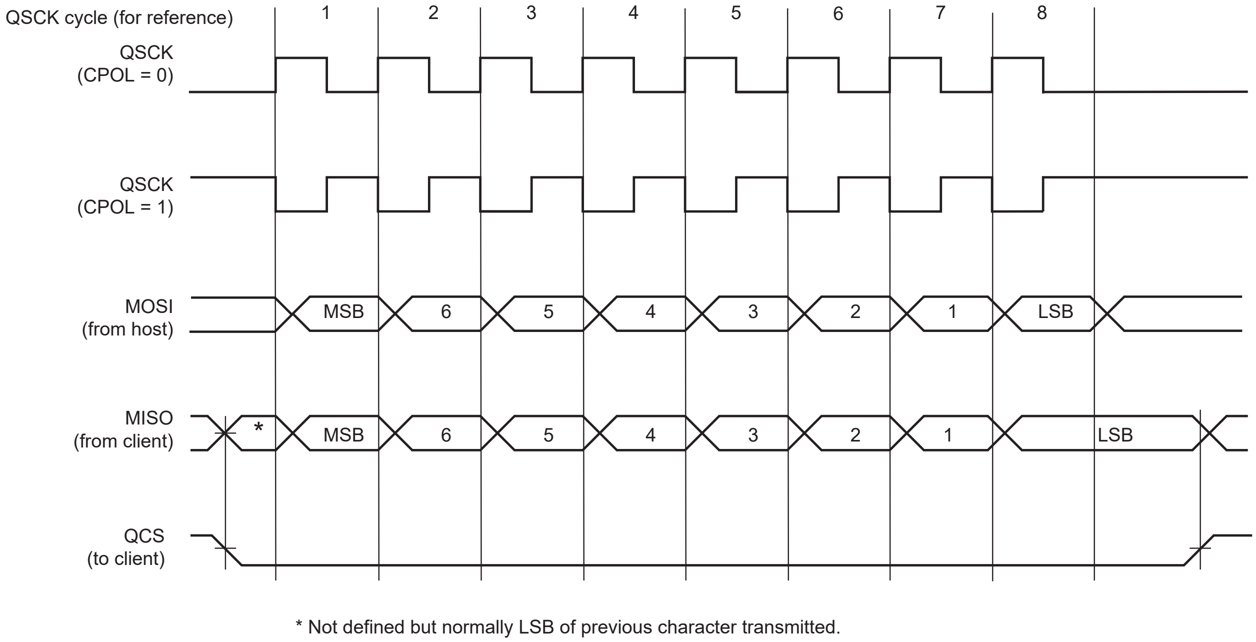
The figures below show examples of data transfers.

Figure 64-4. QSPI Transfer Format (QSPI_SCR.CPHA = 0, 8 bits per transfer)



* Not defined, but normally MSB of previous character received.

Figure 64-5. QSPI Transfer Format (QSPI_SCR.CPHA = 1, 8 bits per transfer)



64.6.4 Transfer Delays

The figure below shows several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the deactivation and the activation of QCS, programmed by writing QSPI_MR.DLYCS. Allows to adjust the minimum time of QCS at high level.
- The delay before QSCK, programmed by writing QSPI_SR.DLYBS. Allows the start of QSCK to be delayed after the chip select has been asserted.
- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT.
  - If QSPI_MR.SMM = 0. Allows insertion of a delay between two consecutive transfers.
  - If QSPI_MR.SMM = 1. Allows insertion of a delay between last QSCK pulse and QCS rise.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 64-6. Programmable Delays (SMM = 0)

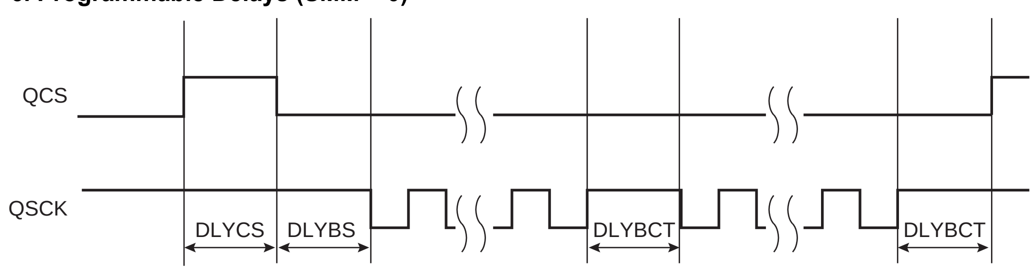
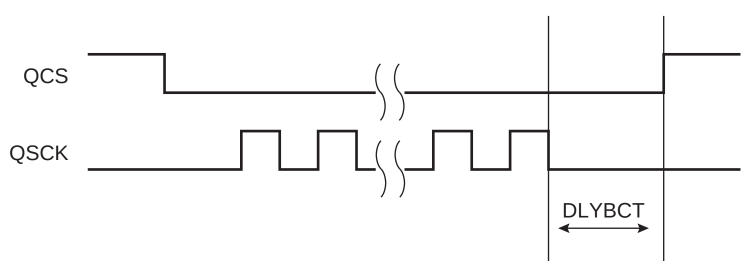


Figure 64-7. DLYBCT with SMM = 1



64.6.5 DLL

The QSPI relies on a DLL to generate the clock required for the Controller core and the physical interface.

To enable the DLL, set QSPI_CR.DLLON then wait for QSPI_SR.DLOCK to be at '1'. Once the DLL is locked, the QSPI Controller core and QSPI physical interface receive the required clocks and can be used.

To disable the DLL, set QSPI_CR.DLLOFF then wait for QSPI_SR.DLOCK to be at '0'. If the DLL is unlocked, the QSPI controller core and QSPI physical interface do not receive the required clocks, power consumption is reduced and the QSPI cannot be used.

Note: The DLL state (ON or OFF) is not affected by a software reset (QSPI_CR.SWRST).

64.6.6 DQS Delay

The DQS Delay analog cell is used to delay the DQS signal in order to use it for data sampling. It is mandatory to enable the DQS Delay cell if QSPI_IFR.DQSEN is set to '1'.

To enable the DQS Delay cell, set QSPI_MR.DQSDLYEN to '1'. This must be done before enabling the QSPI.

Periodically, the DQS Delay cell must be calibrated. The calibration can be performed automatically or manually through a refresh. The calibration process is described in the section Refresh Sequence.

64.6.7 Pad Calibration

The need for output impedance calibration arises with higher data rates. As the data rate increases, some transmission line effects may occur and lead to the generation of undershoots and overshoots, hence degrading the signal quality.

To avoid these transmission problems, a pad calibration cell is used to adjust the output impedance to the driven I/Os.

The pad calibration sequence can be started manually by writing a '1' to QSPI_CR.STPCAL. Then QSPI_SR.CALBSY indicates the state of the I/O calibration. It is advised to run manual pad calibration while the QSPI is disabled (QSPI_SR.QSPIENS = '0') to avoid conflicts with the automatic refresh sequence.

The pad calibration sequence can also be performed automatically through a refresh. See Refresh Sequence.

The pad calibration cell requires a start-up time defined by QSPI_PCALCFG.CALCNT. The default value of this field is adequate for start-up. A different value may be configured in QSPI_PCALCFG.CALCNT.

If QSPI_PCALCFG.AAON is set to '1', the analog circuitry is not shut down at the end of the calibration and the start-up time is only required for the first calibration sequence, thus increasing performance. If QSPI_PCALCFG.AAON is set to '0', the analog circuitry is shut down at the end of the calibration sequence and power consumption is reduced. In this case, the start-up time is performed each time a calibration sequence is started and performance is reduced.

Automatic I/O calibration during the refresh sequence is disabled by writing QSPI_PCALCFG.DAPCAL to '1'. For more details, see Refresh Sequence.

Notes:
1. QSPI_PCALCFG must not be configured while the QSPI is enabled.
2. QSPI_PCALCFG and QSPI_PCALBP are not affected by a software reset (QSPI_CR.SWRST=1).

64.6.8 Refresh Sequence

The QSPI integrates a refresh sequence for the analog blocks to ensure optimal behavior and performance.

Analog blocks which are refreshed during the refresh sequence are:

- DQS delay
- Pad calibration

64.6.8.1 Automatic Refresh

QSPI_REFRESH defines the periodicity of the automatic refresh sequence.

Automatic refresh is enabled if the following conditions are met:

- The QSPI is enabled and QSPI_CR.QSPIENS is written to '1'.
- QSPI_REFRESH.REFRESH is not equal to '0'.

- One additional condition listed below is met:
  - QSPI_MR.DQSDLYEN is set
  - QSPI_PCALCFG.DAPCAL is not set

An automatic refresh sequence of analog blocks is performed:

- after the first QSPI enable
- when the internal refresh counter equals ‘0‘ if QSPI_REFRESH.REFRESH is not equal to ‘0’

64.6.8.2 On-Demand Refresh

A refresh sequence can be launched any time by writing QSPI_CR.STPCAL to ‘1’. This minimizes the latency of the first access as the first calibration is always longer than the one generated by a refresh. To avoid conflicts with an automatic refresh, automatic refresh must be disabled, or the on-demand refresh process should be started only when the QSPI is disabled. The QSPI_ISR.RFRSHD flag indicates the end of the refresh process.

64.6.9 QSPI SPI Mode

In SPI mode, the QSPI acts as a standard SPI Host.

To activate this mode, QSPI_MR.SMM must be written to ‘0’.

64.6.9.1 SPI Mode Operations

The QSPI in standard SPI mode operates on the GCK clock. It fully controls the data transfers to and from the client connected to the SPI bus. The QSPI drives the chip select line to the client (QCS) and the serial clock signal (QSCK).

The QSPI features two holding registers, the Transmit Data register (QSPI_TDR) and the Receive Data register (QSPI_RDR), and a single internal shift register. The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to QSPI_TDR. The written data is immediately transferred to the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted to the internal shift register. Receiving data cannot occur without transmitting data. If receiving mode is not needed, for example when communicating with a client receiver only (such as an LCD), the receive status flags in the Status register (QSPI_ISR) can be discarded.

If new data is written in QSPI_TDR during the transfer, it is retained there until the current transfer is completed. Then, the received data is transferred from the internal shift register to QSPI_RDR, the data in QSPI_TDR is loaded in the internal shift register and a new transfer starts.

The transfer of a data written in QSPI_TDR in the internal shift register is indicated by the Transmit Data Register Empty (TDRE) bit in QSPI_ISR. When new data is written in QSPI_TDR, this bit is cleared. QSPI_ISR.TDRE is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the TXEMPTY flag in QSPI_ISR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, QSPI_ISR.TXEMPTY is set after the completion of this delay. The peripheral clock and GCK clock can be switched off at this time.

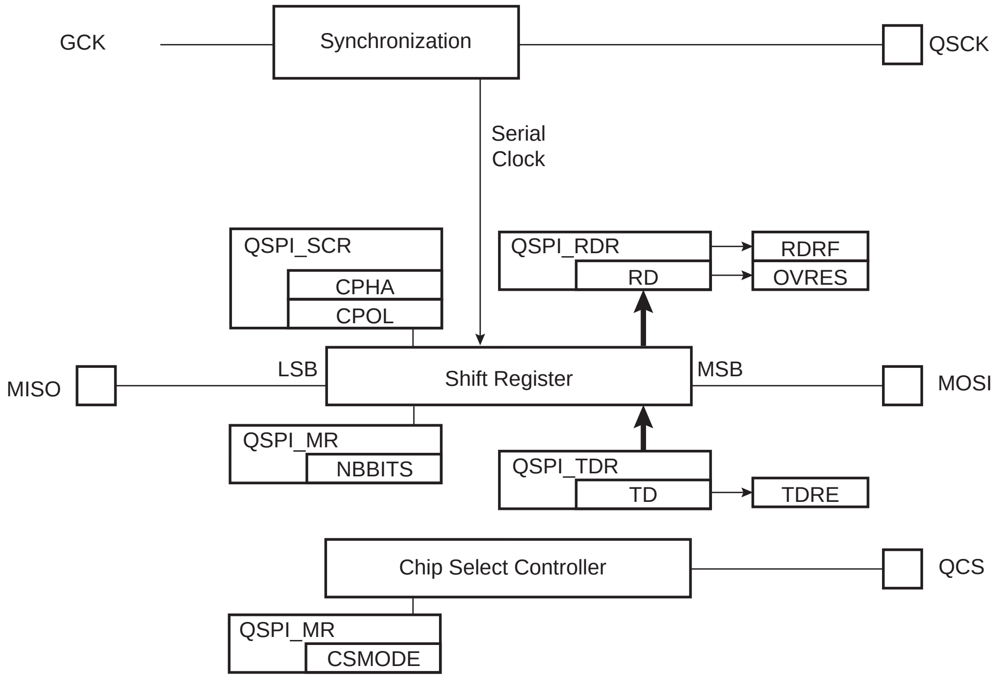
The transfer of received data from the internal shift register in QSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in QSPI_ISR. When the received data is read, the QSPI_ISR.RDRF bit is cleared.

If QSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in QSPI_ISR is set. As long as this flag is set, new data will overwrite the old QSPI_RDR.RD value. The user must read QSPI_ISR to clear the OVRES bit.

SPI Mode Block Diagram shows a block diagram of the SPI when operating in Host mode. SPI Mode Flow Diagram shows a flow chart describing how transfers are handled.

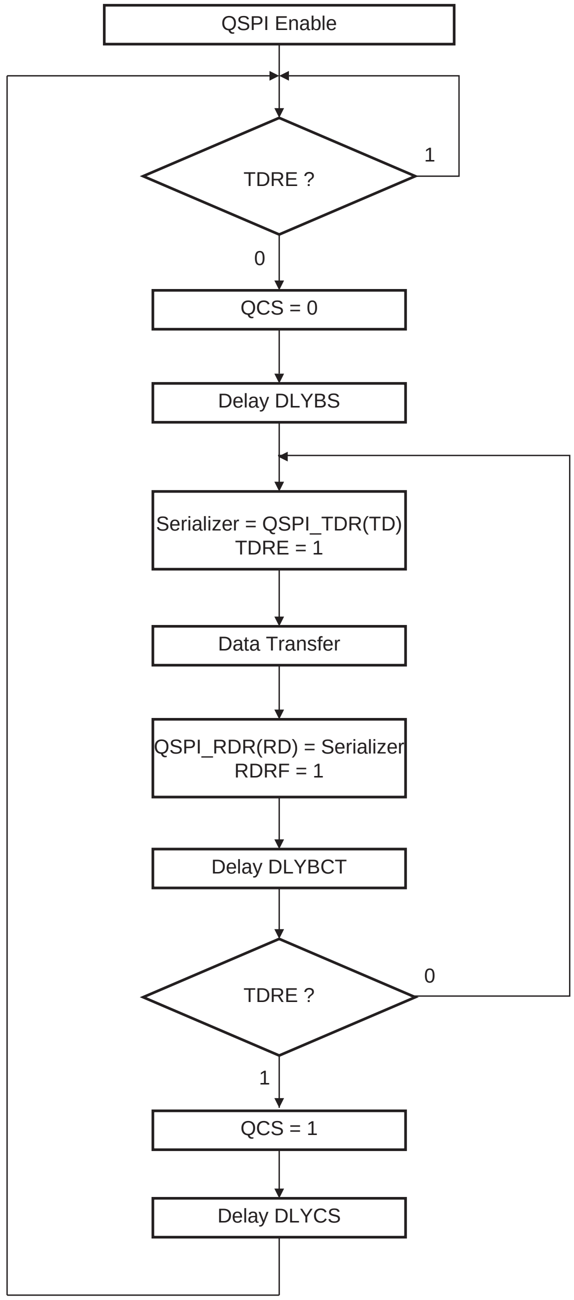
64.6.9.2 SPI Mode Block Diagram

Figure 64-8. SPI Mode Block Diagram



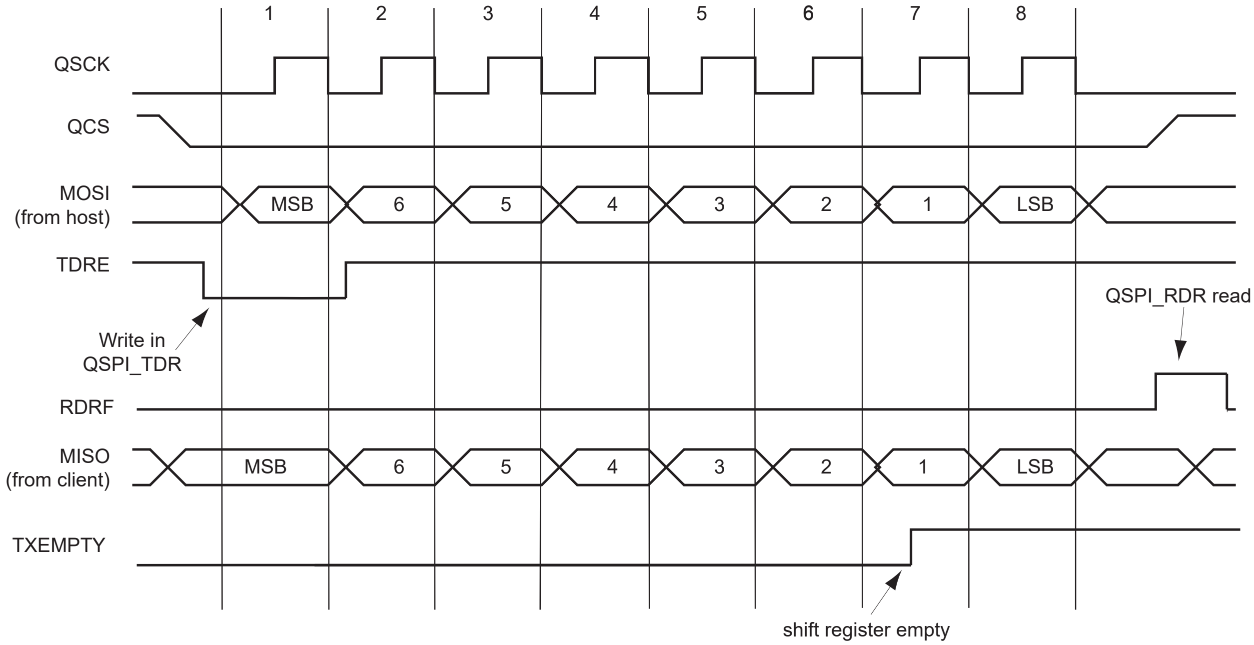
64.6.9.3 SPI Mode Flow Diagram

Figure 64-9. SPI Mode Flow Diagram



The figure below shows Transmit Data Register Empty (TDRE), Receive Data Register Full (RDRF) and Transmission Register Empty (TXEMPTY) status flags behavior within the QSPI_SR during an 8-bit data transfer in Fixed mode, without DMA.

Figure 64-10. Status Register Flags Behavior



Notes: Due to the internal architecture (see Block Diagram):
- A latency occurs between the TXEMPTY rise and the end of the frame.
- A delay occurs between the end of the frame and the RDRF flag rise.

64.6.9.4 Peripheral Deselection without DMA

During a transfer of more than one data on a Chip Select without the DMA, the QSPI_TDR is loaded by the processor and the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shift register. When this flag is detected high, the QSPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, the Chip Select is not deasserted between the two transfers. Depending on the application software handling the QSPI_SR flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload the QSPI_TDR in time to keep the chip select active (low). A null Delay Between Consecutive Transfer (DLYBCT) value in the QSPI_MR gives even less time for the processor to reload the QSPI_TDR. With some SPI client peripherals, requiring the chip select line to remain active (low) during a full set of transfers may lead to communication errors.

To facilitate interfacing with such devices, QSPI_MR.CSMODE may be configured to ‘1’. This allows the chip select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer (LASTXFER) bit in the Control register (QSPI_CR). Even if the QSPI_TDR is not reloaded, the chip select remains active. To have the chip select line rise at the end of the last data transfer, QSPI_CR.LASTXFER must be written to ‘1’ at the same time or after writing the last data to transmit into the QSPI_TDR.

64.6.9.5 Peripheral Deselection with DMA

When the DMA Controller is used, the Chip Select line remains low during the transfer since the TDRE flag is managed by the DMA itself. Reloading the QSPI_TDR by the DMA is done as soon as the TDRE flag is set. In this case, writing QSPI_MR.CSMODE to ‘1’ may not be needed. However, when other DMA channels connected to other peripherals are also in use, the QSPI DMA could be delayed by another DMA with a higher priority on the bus. Having DMA buffers in slower memories like Flash memory or SDRAM compared to fast internal SRAM, may lengthen the reload time of the QSPI_TDR by the DMA as well. This means that the QSPI_TDR might not be reloaded in time to keep the chip select line low. In this case, the chip select line may toggle between data transfer and according to some SPI client devices, the communication might get lost. It may be necessary to configure QSPI_MR.CSMODE to ‘1’.

When QSPI_MR.CSMODE is configured to ‘0’, the QCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shifter. When this flag is detected, the QSPI_TDR can be reloaded. If this reload occurs before the end of the current transfer, the Chip Select is not deasserted between the two transfers. This might lead to difficulties for interfacing with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices, the QSPI_MR may be configured with QSPI_MR.CSMODE at ‘2’.

64.6.10 QSPI Serial Memory Mode

In Serial Memory mode, the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock, etc.) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual SPI, Quad and Octal SPI protocols.

To activate this mode, QSPI_MR.SMM must be written to ‘1’.

In Serial Memory mode, data is transferred either by QSPI_TDR and QSPI_RDR or by writing or read in the QSPI memory space (0x20000000–0x30000000) depending on QSPI_IFR.TFRTYP and QSPI_IFR.SMRM configuration.

64.6.10.1 Initialization

The QSPI initialization procedure must follow the specific steps below to ensure proper behavior.

1. Configure QSPI_PCALCFG.
2. Write:
   – QSPI_CR. DLLON
   – QSPI_CR.STPCAL
3. Wait for:
   – QSPI_SR.DLOCK=1
   – QSPI_SR.CALBSY=0
4. Configure QSPI_REFRESH, QSPI_MR, QSPI_SCR, QSPI_SMR, etc.
5. Wait for QSPI_SR.SYNCBSY = 0, then write QSPI_CR.UPDCFG. Wait for QSPI_SR.SYNCBSY = 0.
6. Write QSPI_CR.QSPIEN.
7. Wait for QSPI_SR.QSPIENS = 1.
8. If automatic refresh is enabled, wait for the flag QSPI_SR.RFRSHD to rise by polling or interrupt.

64.6.10.2 Suspend

The QSPI suspend procedure must follow some specific state to ensure proper behavior.

1. Wait for QSPI_SR.RBUSY = 0 and QSPI_SR.HIDLE = 1.
2. Set QSPI_CR.QSPIDIS and wait for QSPI_SR.QSPIENS = 0.
3. Disable GCK clock.
4. Set QSPI_CR.DLLOFF and wait for QSPI_SR.DLOCK = 0.
5. Wait for QSPI_SR.CALBSY = 0.
6. Disable the peripheral clock.

64.6.10.3 Instruction Frame

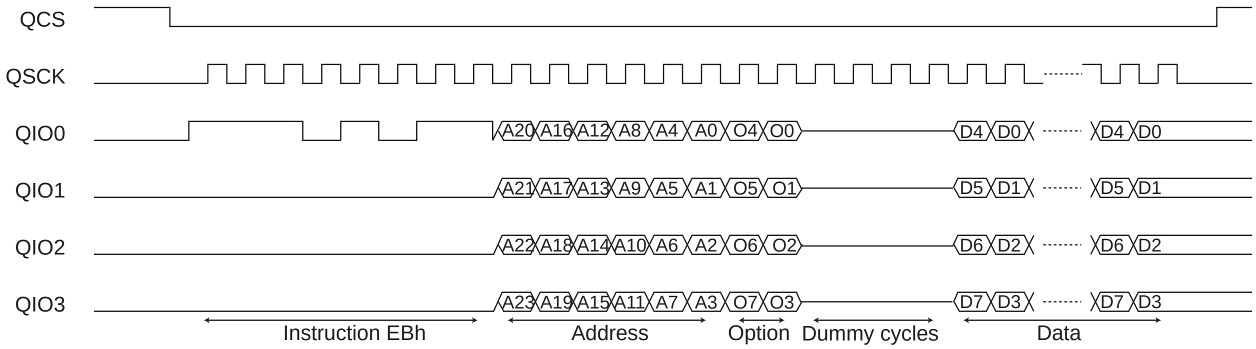
In order to control serial Flash memories, the QSPI is able to send instructions via the SPI bus (e.g. READ, PROGRAM, ERASE, LOCK, etc.). Because the instruction set implemented in serial Flash memories is memory-vendor dependent, the QSPI includes a complete Instruction Frame register (QSPI_IFR) to ensure compatibility with all serial Flash memories.

An instruction frame includes:

- An instruction code. The instruction is optional in some cases (see Continuous Read Mode).
- An address (size: 8, 16, 24 or 32 bits). The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default, the address is 8 bits long, but can be increased up to 32 bits to support serial Flash memories larger than 128 Mbits (16 Mbytes).
- An option code (size: 1/2/4/8 bits). The option code is used to activate the XIP mode or the Continuous Read mode (see Continuous Read Mode) for READ instructions, in some serial Flash memory devices. These modes improve the data read latency.
- Dummy cycles. Dummy cycles are optional but required by some READ instructions.
- Data bytes are optional. Data bytes are present for data transfer instructions such as READ or PROGRAM.

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.

Figure 64-11. Instruction Frame



64.6.10.4 Instruction Frame Transmission

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the fields WRINST, WROPT, RDINST and RDOPT in the Write Instruction Code register (QSPI_WICR) and the Read Instruction Code register (QSPI_RICR). QSPI_WICR configures instruction code and option code for write accesses, and QSPI_RICR configures instruction code and option code for read accesses. If a frame is without data (QSPI_IFR.DATAEN = 0), then QSPI_WICR is used for instruction code and option code.

Then, the user must write QSPI_IFR to configure the instruction frame depending on which instruction must be sent.

The instruction frame is configured by the following bits and fields of QSPI_IFR:

- WIDTH field—used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data.
- INSTEN bit—used to enable the send of an instruction code.
- ADDREN bit—used to enable the send of an address after the instruction code.
- OPTEN bit—used to enable the send of an option code after the address.
- DATAEN bit—used to enable the transfer of data (READ or PROGRAM instruction).
- OPTL field—used to configure the option code length. The value written in OPTL must be consistent with the value written in the field WIDTH. For example: OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).
- ADDRL bit—used to configure the address length.
- TFRTYP field—used to define which type of data transfer must be performed.
- CRM bit—used to enable the continuous read mode, see Continuous Read Mode.
- DDREN bit—used to configure the Double Data Rate mode; instruction code is still transmitted in Single Data Rate mode. Instruction code can be transmitted in DDR mode by writing a ‘1’ to QSPI_IFR.DDRCMDEN.
- NBDUM field—used to configure the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.
- END bit—used to define the endianness of the targeted memory.
- SMRM bit—when TFRTYP = ‘0’, defines if instruction frame transmission is triggered by register accesses or QSPI memory space accesses.
- APBTFRTYP bit—used to define the APB register transfer to memory type (read or write) when QSPI_IFR.TFRTYP is written to ‘0’.
- DQSEN bit—used to define if the targeted memory supplies a DQS signal.
- DDRCMDEN bit—used to define if the instruction code must be sent in DDR mode when QSPI_IFR.DDREN bit is written to ‘1’.
- HFWBEN field—used to enable the HyperFlash Write Buffer command support. In this mode, a new command is generated for each write access. See HyperFlash Mode.
- PROTTYP bit—used to define the QSPI protocol type.

See QSPI Instruction Frame Register.

For instruction frame transmission, memory array accesses are different from the memory register/command accesses, as well as from accesses using the peripheral bus interface or the system bus interface.

Table 64-3. QSPI Access Triggering

QSPI_IFR Configuration	Accesses Triggered by Performing QSPI Memory Space Access	Accesses Triggered by QSPI Register Accesses
QSPI_IFR.TFRTYP = 0 and QSPI_IFR.SMRM = 1	No	Yes
QSPI_IFR.TFRTYP = 0 and QSPI_IFR.SMRM = 0	Yes	No
QSPI_IFR.TFRTYP = 1 or QSPI_IFR.SMRM = 0	Yes	No

64.6.10.4.1 Memory Registers/Commands Access

To perform memory register/command accesses, QSPI_IFR.TFRTYP must be set to ‘0’.

If the frame does not contain any data (e.g WRITE ENABLE command) or if QSPI_IFR.SMRM is set to ‘1’, the user must first configure the address to send by writing QSPI_IAR.ADDR (in the case the frame contains an address field).

- SMRM = 1

When QSPI_IFR.SMRM is set to ‘1’, accesses to the memory are triggered and controlled by QSPI registers.

QSPI_IAR.ADDR must be configured if the frame contains an address field.

QSPI_IFR.APBTFRTYP is used to define whether the access is a read access or a write access. Write frames are triggered by writing QSPI_TDR and read frames are triggered by setting QSPI_CR.STTFR. Each time a new transfer trigger is issued, an SPI transfer is performed with a byte size or halfword size if QSPI_IFR.WIDTH field is configured to either OCT_OUTPUT, OCT_IO or OCT_CMD and QSPI_IFR.DDREN=1. Another byte or halfword is read each time QSPI_RDR is read (flag RDRF shows when a data can be read in QSPI_RDR) or written each time QSPI_TDR is written (flag TDRE shows when a new data can be written). The SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Register Access) for details.

- SMRM = 0

When QSPI_IFR.SMRM is set to ‘0’, accesses to the memory are triggered by performing an access in the QSPI memory space. The address of the instruction frame is defined by the address of the first data access in the QSPI memory space. The addresses of the next accesses are not used by the QSPI.

QSPI_WRACNT.NBWRA defines the number of bytes to be sent to the memory before QSPI_ISR.LWRA rises, which indicates that the last byte has been transmitted. If QSPI_IFR.PROTTYP = ‘3’ and QSPI_IFR.HFWBEN = ‘1’, QSPI_WRACNT.NBWRA is reset after setting QSPI_IFR.HFWBEN bit to ‘1’ and counts data on every frame. At each system bus access, an SPI transfer is performed with the same size. For example, a halfword system bus access leads to a 16-bit SPI transfer, and a byte system bus access leads to an 8-bit SPI transfer. A special case is an instruction frame with address field and no data. In this case the instruction frame is triggered by setting QSPI_CR.STTFR and QPI_IAR is used for the address field. See Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access) for details.

64.6.10.4.2 Memory Array Access

To access the QSPI memory array, QSPI_IFR.TFRTYP must be set to ‘1’. In the case of write access to the QSPI memory array, QSPI_IFR.TFRTYP can be set to ‘0’ with QSPI_IFR.SMRM set to ‘1’.

- TFRTYP = 0 and SMRM = 1

This configuration is allowed for write memory array access only (read access to the memory array is not supported in this configuration). When QSPI_IFR.SMRM is set to '1', accesses to the memory are triggered and controlled by QSPI registers. QSPI_IAR.ADDR must be configured with the address of the first data to write if the frame contains an address field, this field is the one used for the instruction frame address field. The QSPI_IFR.APBTFRTYP must be set to '0'. Write frames are triggered writing the QSPI_TDR register. Each time a new transfer trigger is issued, an SPI transfer is performed with a byte size or halfword size if QSPI_IFR.WIDTH field is configured to OCT_OUTPUT, OCT_IO or OCT_CMD and QSPI_IFR.DDREN=1. Another byte or halfword is written each time QSPI_TDR is written (flag TDRE shows when a new data can be written). If a data is not consecutive to the previously sent data, a new frame must be issued. The SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) for details.

- TFRTYP = 1

When QSPI_IFR.TFRTYP is set to '1', accesses to the memory are triggered by performing an access in the QSPI memory space. The address of the instruction frame is defined by the address of the first data access in the QSPI memory space. Each time the accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read/write data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus access for read accesses. QSPI_WRACNT.NBWRA generates a rising flag when a given number of bytes have been sent to the memory. QSPI_ISR.LWRA indicates the transmission of the last byte. The NBWRA internal counter is reset and begins counting after each start of instruction frame except when QSPI_IFR.PROTTYP = '3' and QSPI_IFR.HFWBEN = 1 where the NBWRA counter is reset after setting the QSPI_IFR.HFWBEN bit to '1' and counts data on every frame.

In the case of read accesses to the memory array, QSPI_SR.RBUSY must be at '0' before terminating the frame. The last SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram TFRTYP = 1, Memory Write Access and Instruction Transmission Flow Diagram TFRTYP = 1, Memory Read Access for details.

The following figures illustrate instruction transmission management.

Figure 64-12. Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Register Access)

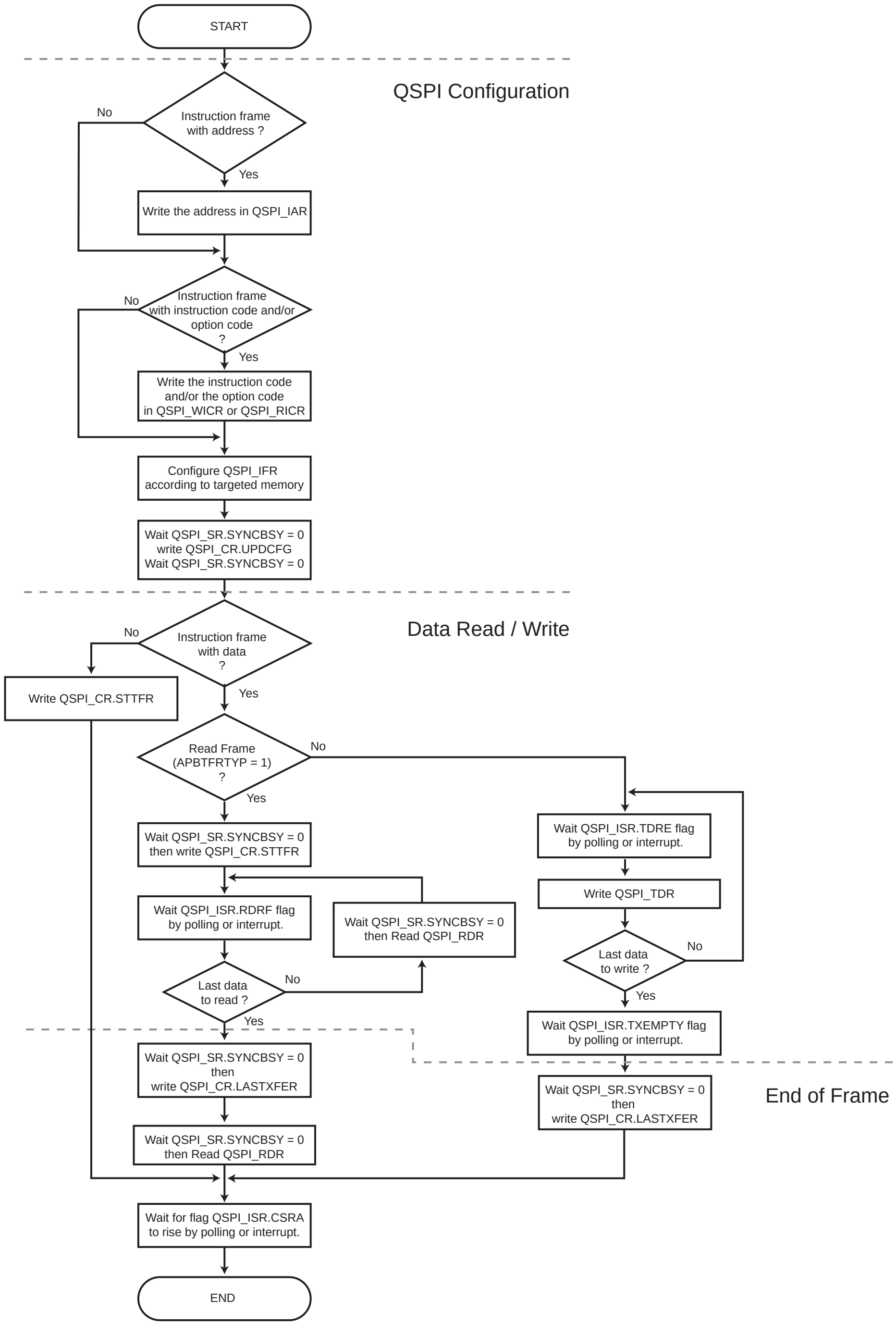


Figure 64-13. Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access)

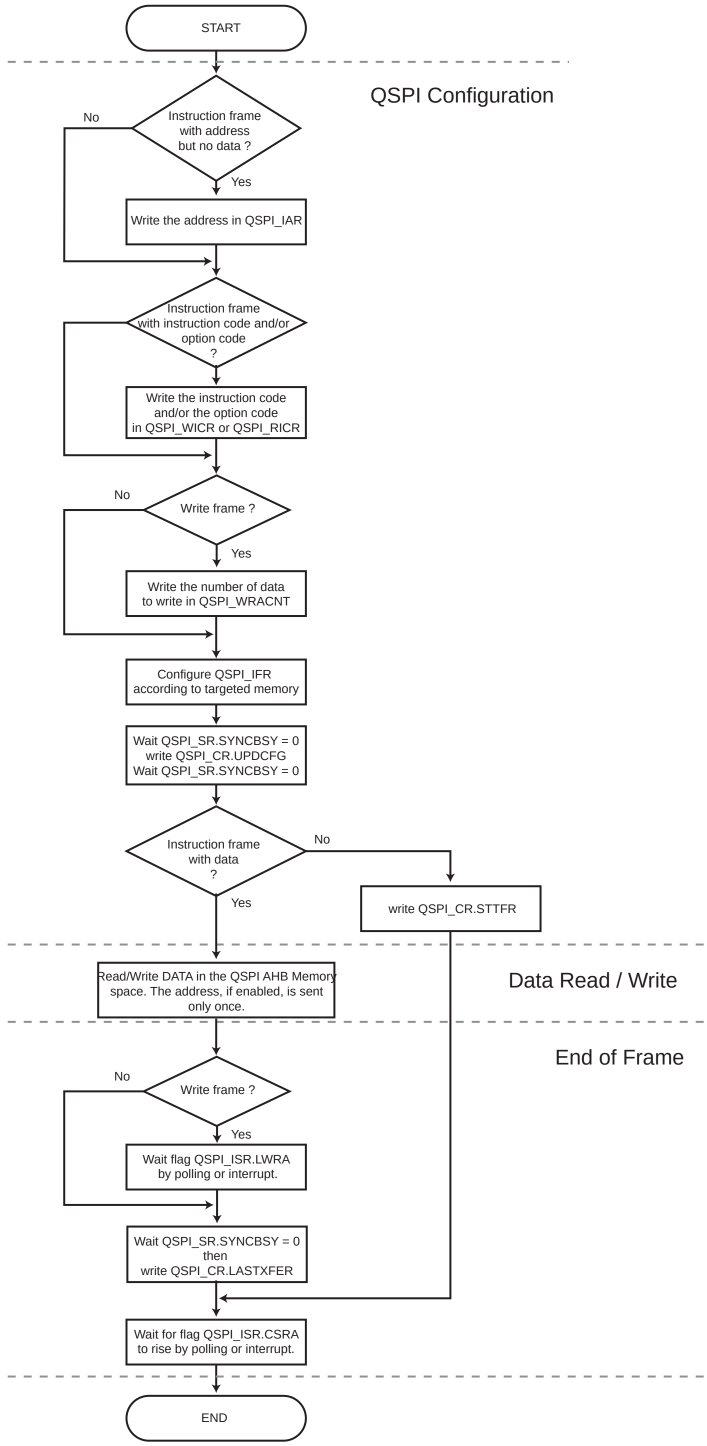


Figure 64-14. Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access)

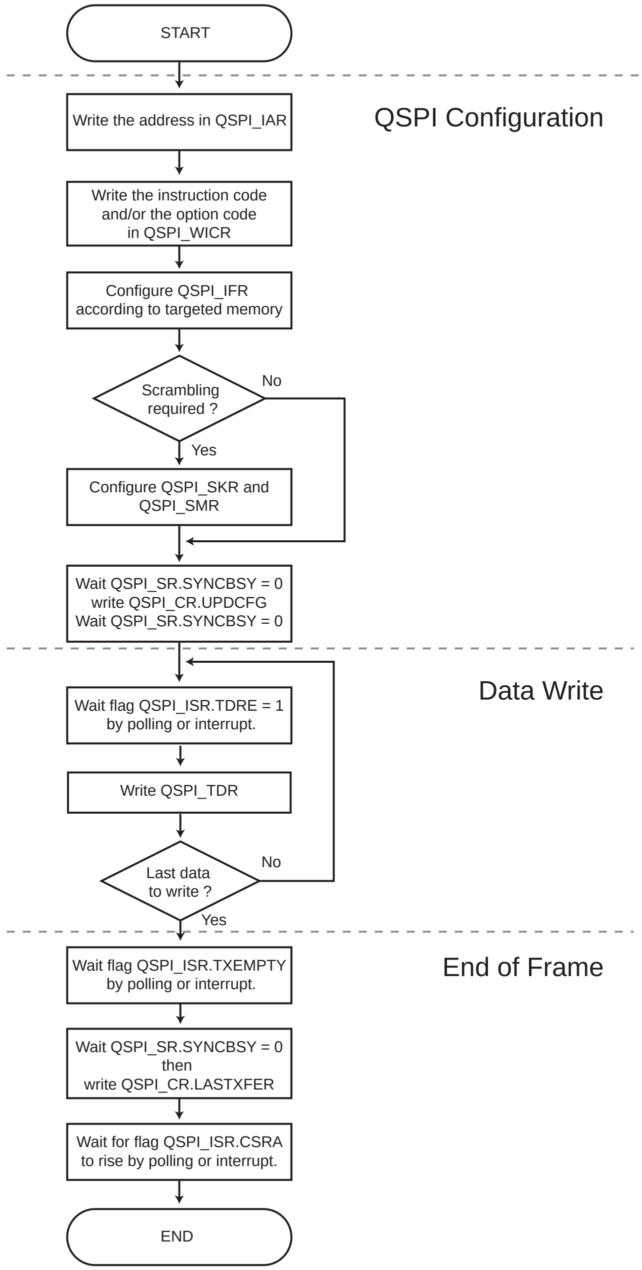
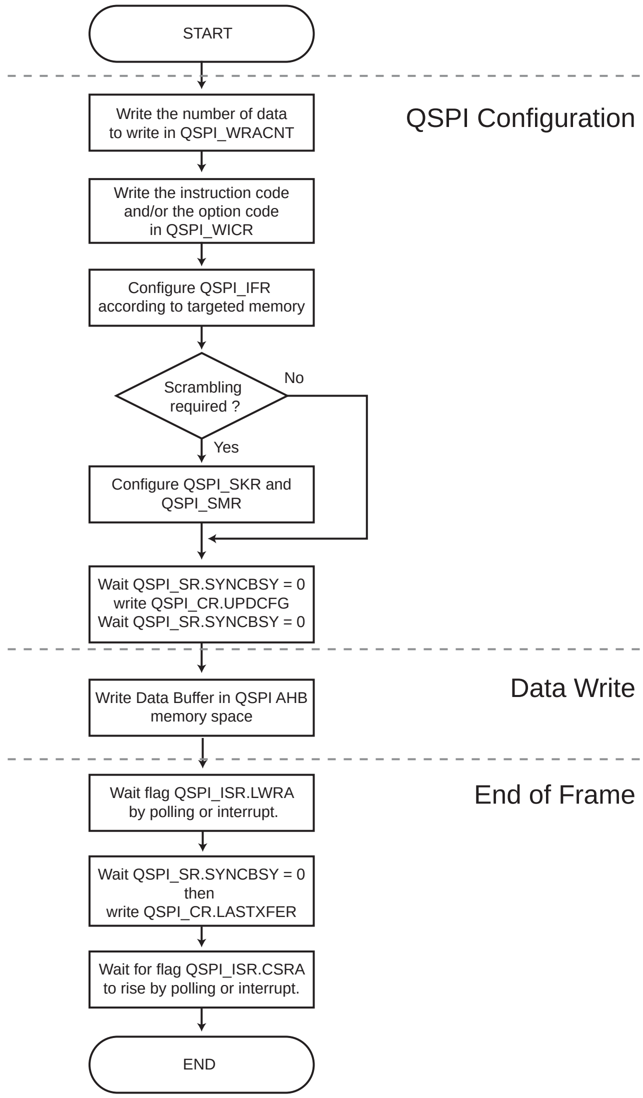


Figure 64-15. Instruction Transmission Flow Diagram TFRTYP = 1, Memory Write Access



Note: For a QSPI Flash memory write, the maximum data buffer size is the Flash page buffer size.

Figure 64-16. Instruction Transmission Flow Diagram TFRTYP = 1, Memory Read Access

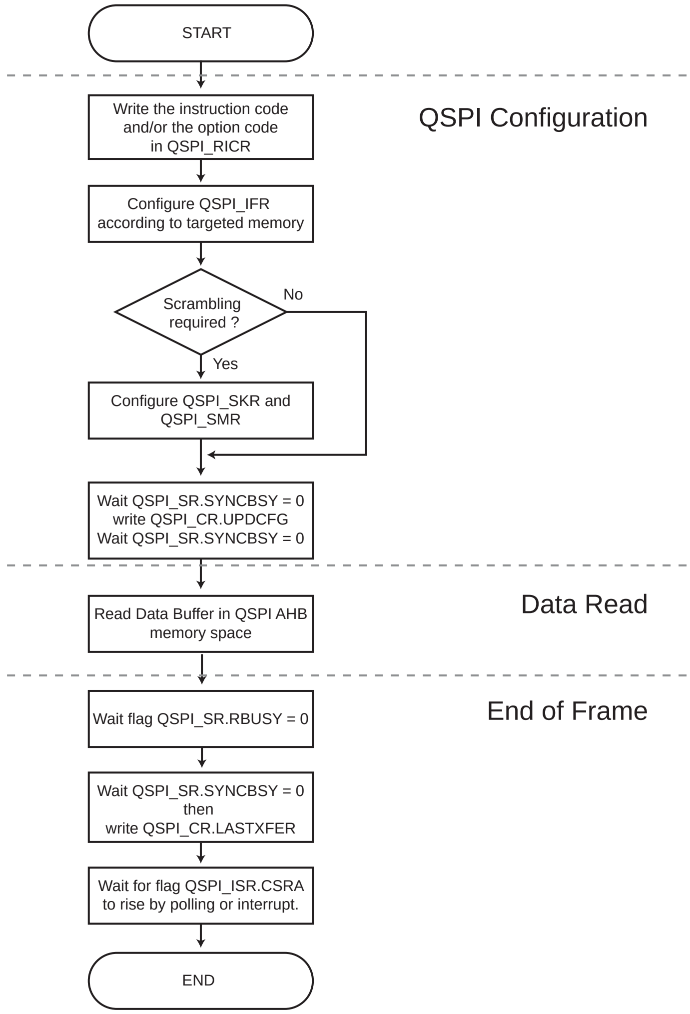


Figure 64-17. HyperFlash “Write Buffer” Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access)

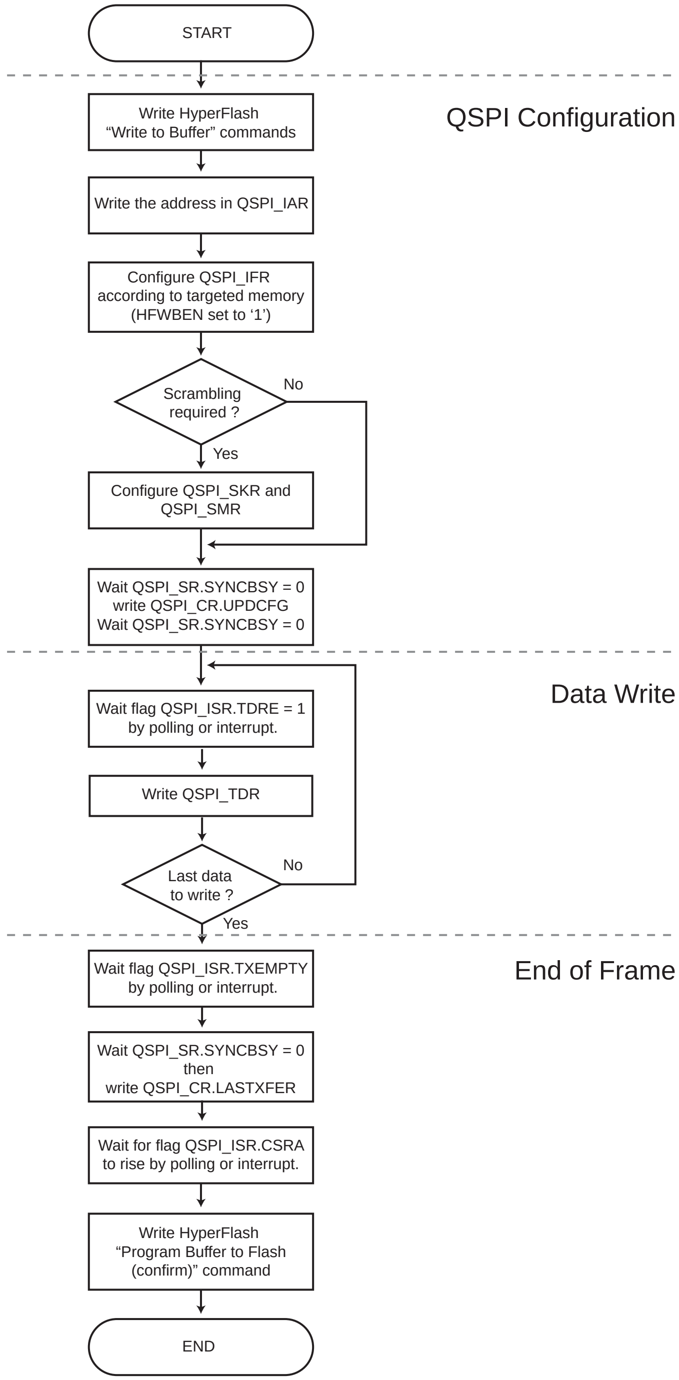
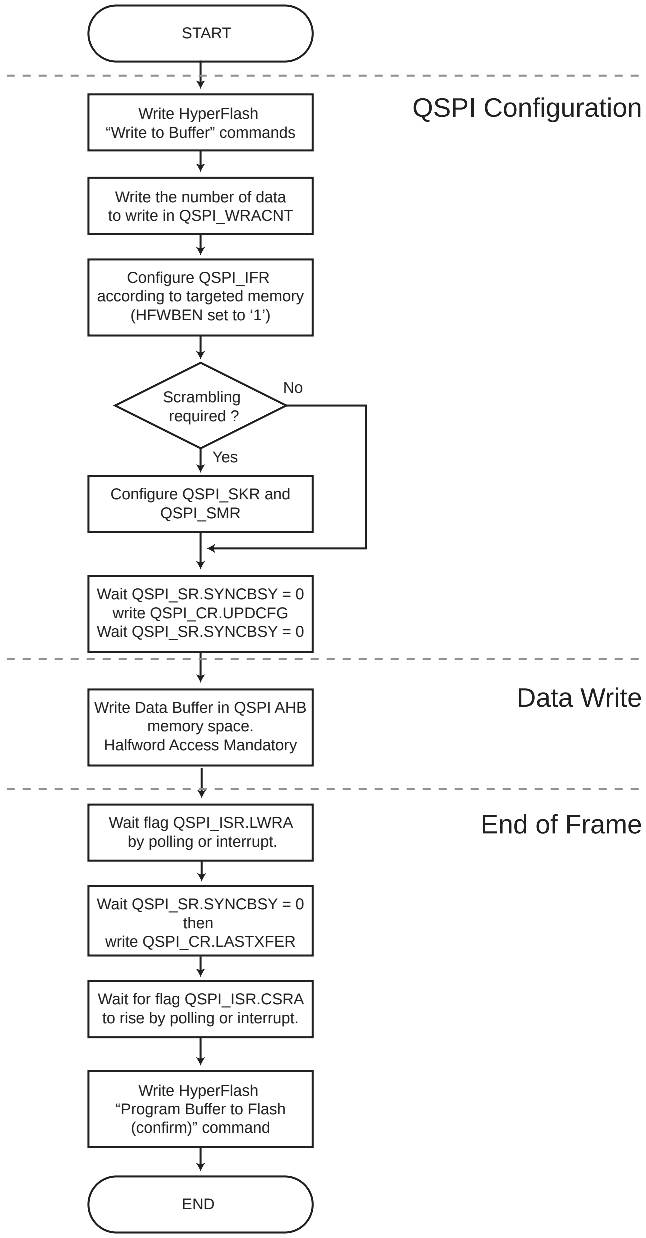


Figure 64-18. HyperFlash “Write Buffer” Flow Diagram TFRTYP = 1, Memory Write Access



64.6.10.5 Write Memory Transfer

Many QSPI memories impose sequential write of the data buffer. When using this type of memory, the device configuration must ensure that the access order is preserved when crossing the different data buses to the QSPI.

64.6.10.6 Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with QSPI_IFR.DATAEN = 1 and QSPI_IFR.TFRTYP = 1.

In this mode, the QSPI is able to read data at random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

In order to fetch data, the user must first configure the instruction frame by writing the QSPI_IFR. Then data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses match the address of the data inside the serial Flash memory.

When Fetch mode is enabled, several instruction frames can be sent before writing QSPI_CR.LASTXFR. Each time the system bus read accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

64.6.10.7 Continuous Read Mode

The QSPI is compatible with the Continuous Read mode which is implemented in some serial Flash memories.

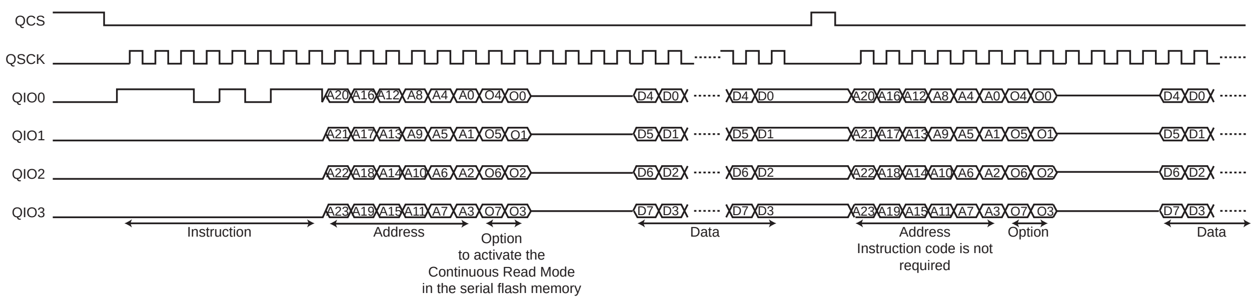
In Continuous Read mode, the instruction overhead is reduced by excluding the instruction code from the instruction frame. When the Continuous Read mode is activated in a serial Flash memory by a specific option code, the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required as the memory uses the stored one.

In the QSPI, Continuous Read mode is used when reading data from the memory (QSPI_IFR.TFRTYP = 1). The addresses of the system bus read accesses are often nonsequential and this leads to many instruction frames that have the same instruction code. By disabling the send of the instruction code, the Continuous Read mode reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the serial Flash memory. The Continuous Read mode is enabled in the QSPI by writing CRM to ‘1’ in the QSPI_IFR (TFRTYP must equal 1). The Continuous Read mode is enabled in the serial Flash memory by sending a specific option code.

⚠ CAUTION If the Continuous Read mode is not supported by the serial Flash memory or disabled, CRM bit must not be written to ‘1’, otherwise data read out of the serial Flash memory is unpredictable.

Figure 64-19. Continuous Read Mode



64.6.10.8 Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (QSPI_SCR.CPOL = 0 and QSPI_SCR.CPHA = 0; see Serial Clock Phase and Polarity).

All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

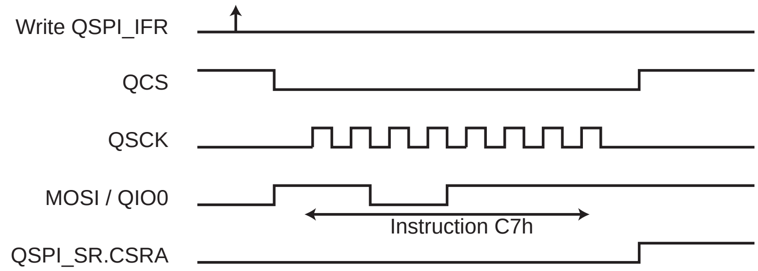
Example 1:

Instruction in Single-bit SPI, without address, without option, without data.

Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 in QSPI_WICR.
- Write 0x0000_0010 in QSPI_IFR.
- Update configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to ‘1’.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-20. Instruction Transmission Waveform 1



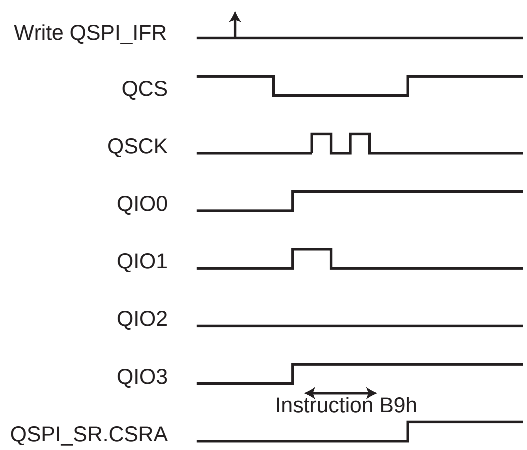
Example 2:

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 in QSPI_WICR.
- Write 0x0000_0016 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to '1'
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-21. Instruction Transmission Waveform 2



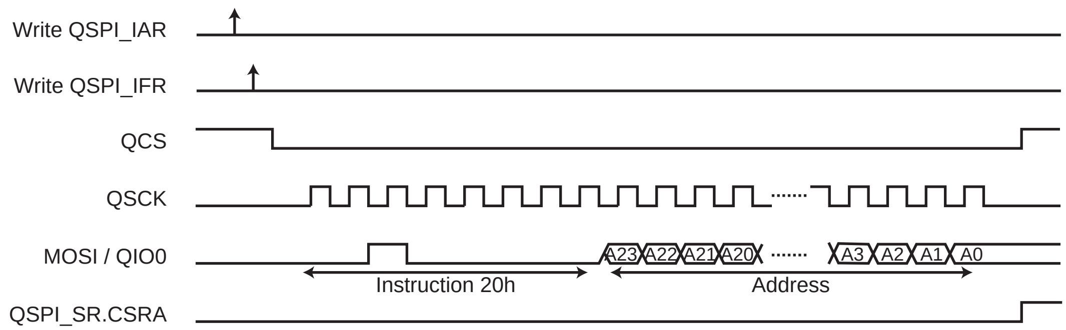
Example 3:

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) in QSPI_IAR.
- Write 0x0000_0020 in QSPI_WICR.
- Write 0x0000_0030 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to '1'
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-22. Instruction Transmission Waveform 3



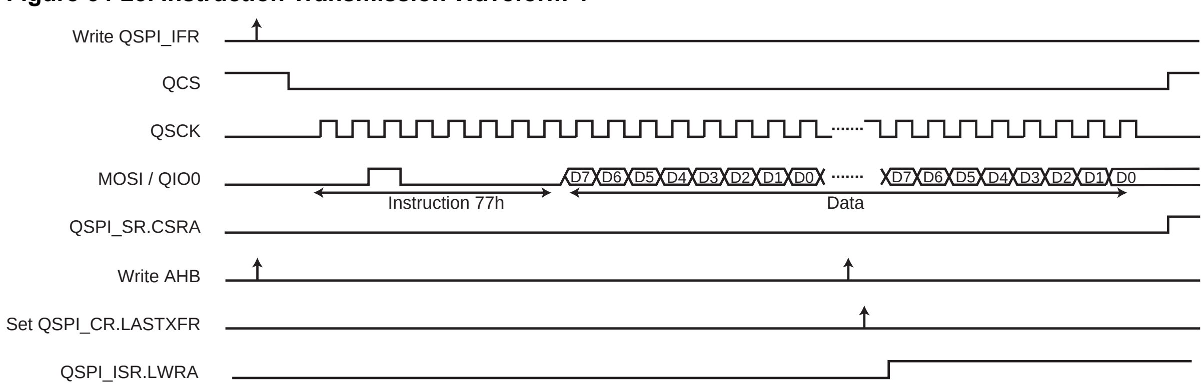
Example 4:

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 in QSPI_WICR.
- Write 0x0000_0090 in QSPI_IFR.
- Write QSPI_WRACNT.NBWRA with the number of bytes to write.
- Update Configuration (See Updating the QSPI Configuration).
- Write data in the system bus memory space (0x20000000–0x30000000). The address of system bus write accesses is not used.
- Wait for QSPI_ISR.LWRA to rise.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-23. Instruction Transmission Waveform 4



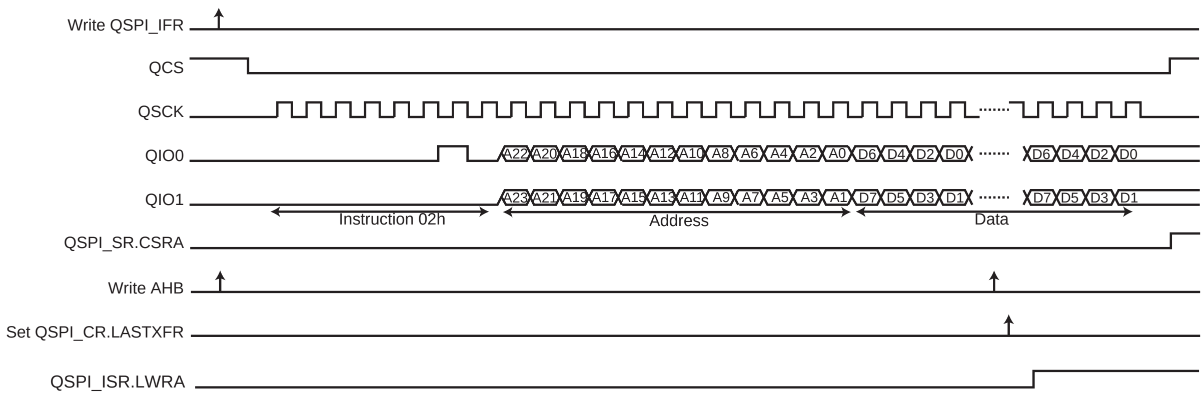
Example 5:

Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 in QSPI_WICR.
- Write 0x0000_18B3 in QSPI_IFR.
- Write QSPI_WRACNT.NBWRA with the number of bytes to write.
- Update Configuration (See Updating the QSPI Configuration).
- Write data in the QSPI system bus memory space (0x20000000–0x30000000). The address of the first system bus write access is sent in the instruction frame. The address of the next system bus write accesses is not used.
- Wait for QSPI_ISR.LWRA to rise.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-24. Instruction Transmission Waveform 5



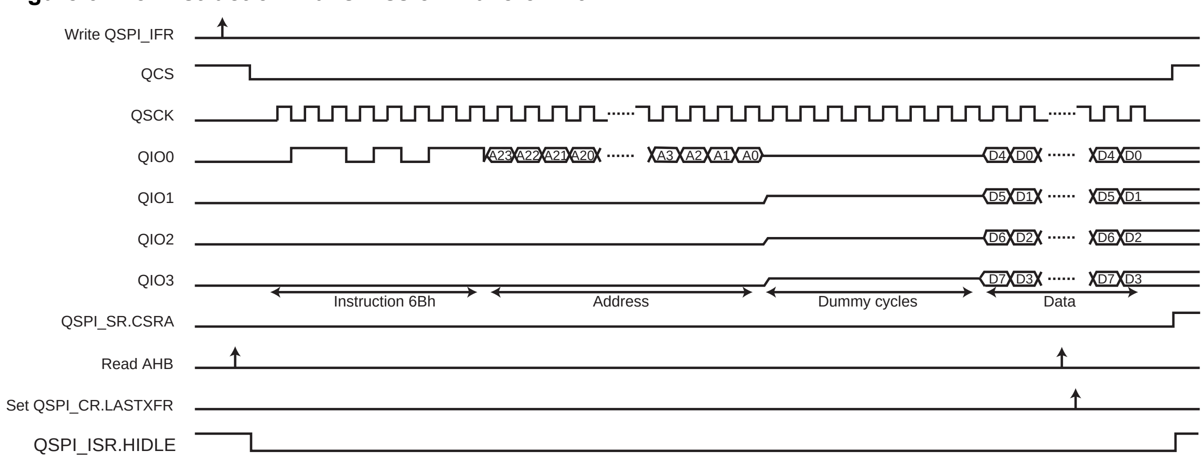
Example 6:

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B in QSPI_RICR.
- Write 0x0008_18B2 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x20000000–0x30000000).
  The address of the first system bus read access is sent in the instruction frame.
  The address of the next system bus read accesses is not used.
- Wait for QSPI_SR.RBUSY to be at ‘0’.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-25. Instruction Transmission Waveform 6



Example 7:

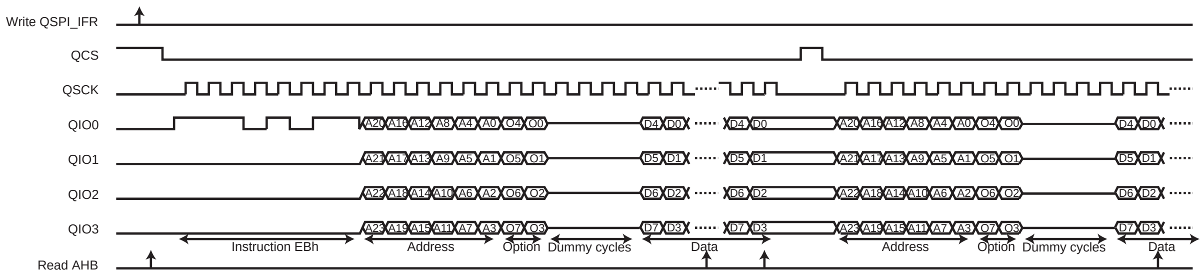
Instruction in Single-bit SPI, with address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_RICR.
- Write 0x0004_3BF4 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).

- Read data in the QSPI system bus memory space (0x20000000–0x30000000). Fetch is enabled, the address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY to be at ‘0’.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-26. Instruction Transmission Waveform 7



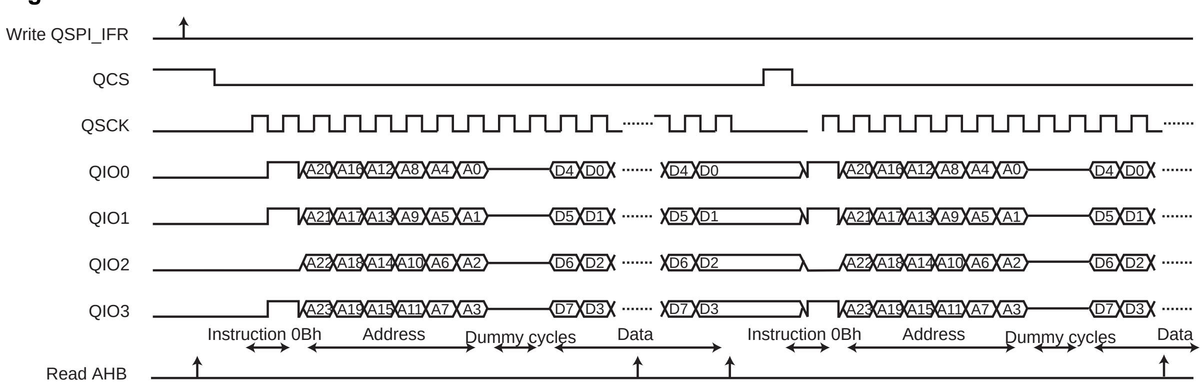
Example 8:

Instruction in Quad SPI, with address in Quad SPI, without option, with data read in Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_RICR.
- Write 0x0002_28B6 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x20000000–0x30000000). Fetch is enabled, the address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY to be at ‘0’.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-27. Instruction Transmission Waveform 8



Example 9:

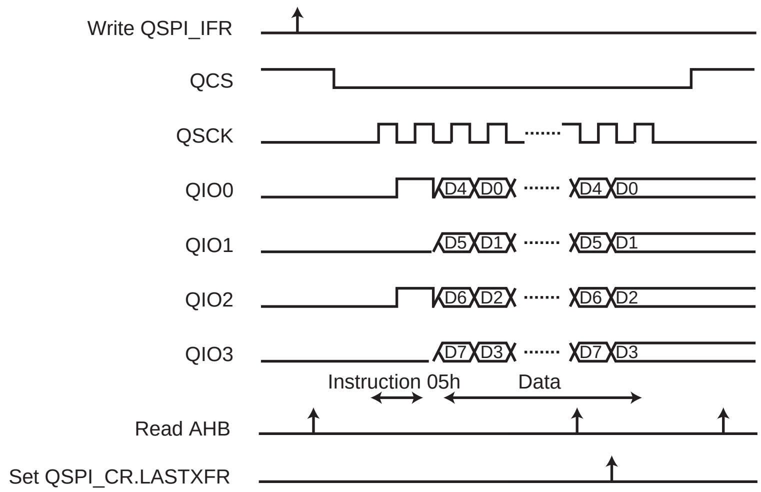
Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch.

Command: HIGH-SPEED READ (05h)

- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0000_0096 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x20000000–0x30000000). Fetch is disabled.
- Wait for QSPI_SR.RBUSY to be at ‘0’.

- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-28. Instruction Transmission Waveform 9



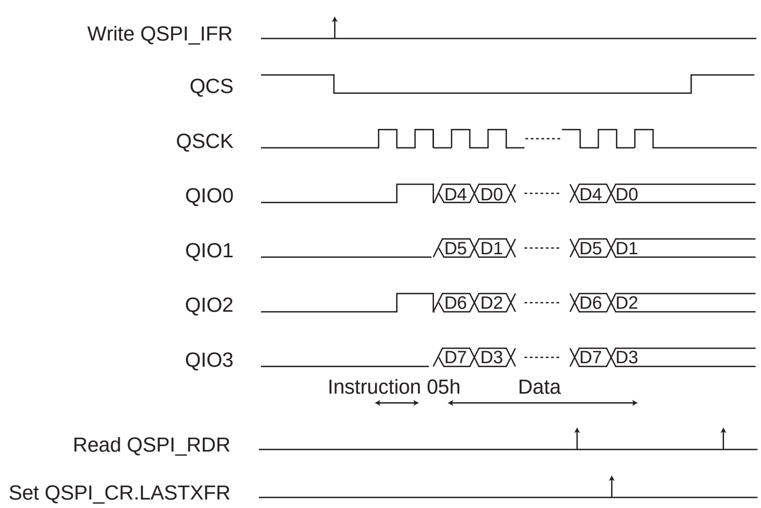
Example 10:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch, read launched through APB interface.

Command: HIGH-SPEED READ (05h)

- Set SMRM to ‘1’ and TFRTYP to ‘0’ in QSPI_MR
- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0100_0096 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Write a ‘1’ to QSPI_CR.STTFR.
- Wait flag RDRF and Read data in the QSPI_RDR register
  Fetch is disabled.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.
- Read data in the QSPI_RDR register.

Figure 64-29. Instruction Transmission Waveform 10



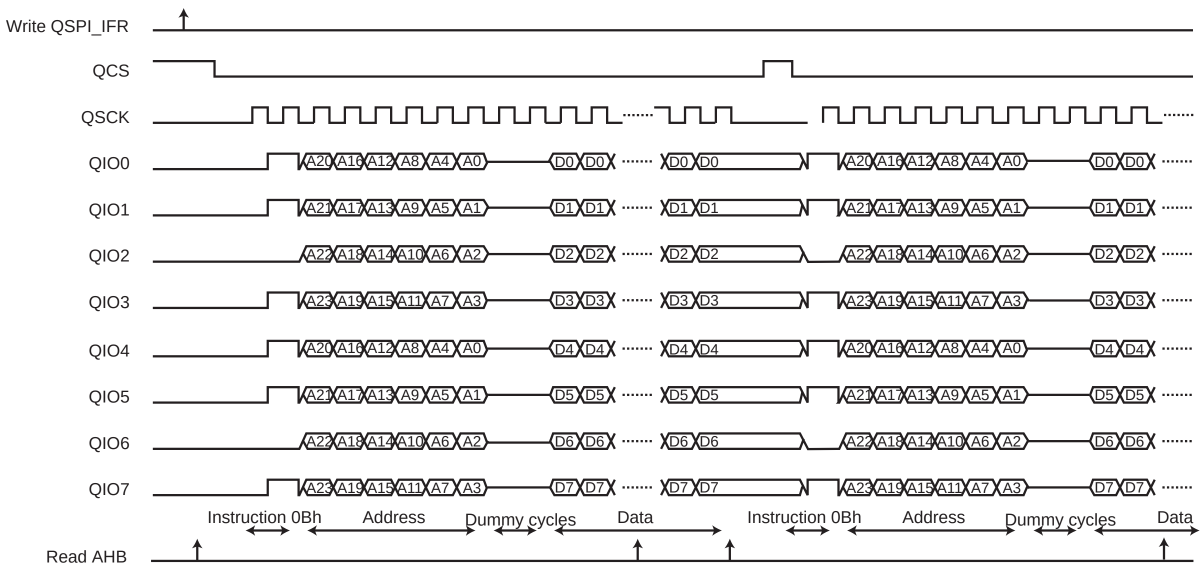
Example 11:

Instruction in Octal Twin-Quad SPI, with address in Octal Twin-Quad SPI, without option, with data read in Octal Twin-Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_RICR.
- Write 0x1002_18B9 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x20000000–0x30000000).
  Fetch is enabled, the address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY to be at ‘0’.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-30. Instruction Transmission Waveform 11



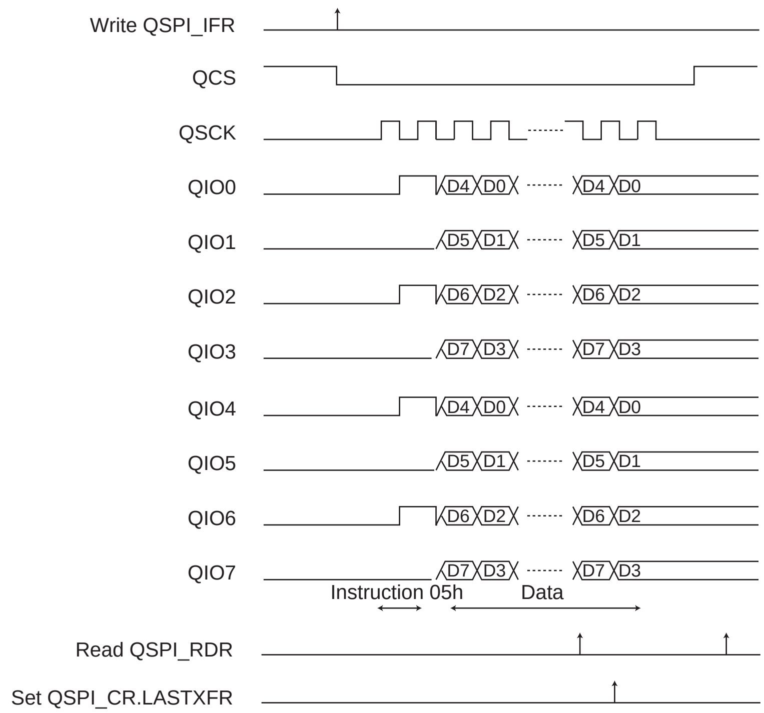
Example 12:

Instruction in Octal Twin-Quad SPI, without address, without option, with data read in Octal Twin-Quad SPI, without dummy cycles, without fetch, read launched through APB interface.

Command: HIGH-SPEED READ (05h)

- Write a ‘1’ to QSPI_MR.SMRM and a ‘0’ to TFRTYP.
- Write 0x0000_0005 in QSPI_RICR.
- Write 0x1100_0099 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Write a ‘1’ to QSPI_CR.STTFR.
- Wait flag RDRF and Read data in the QSPI_RDR register
  Fetch is disabled.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.
- Read data in the QSPI_RDR register.

Figure 64-31. Instruction Transmission Waveform 12



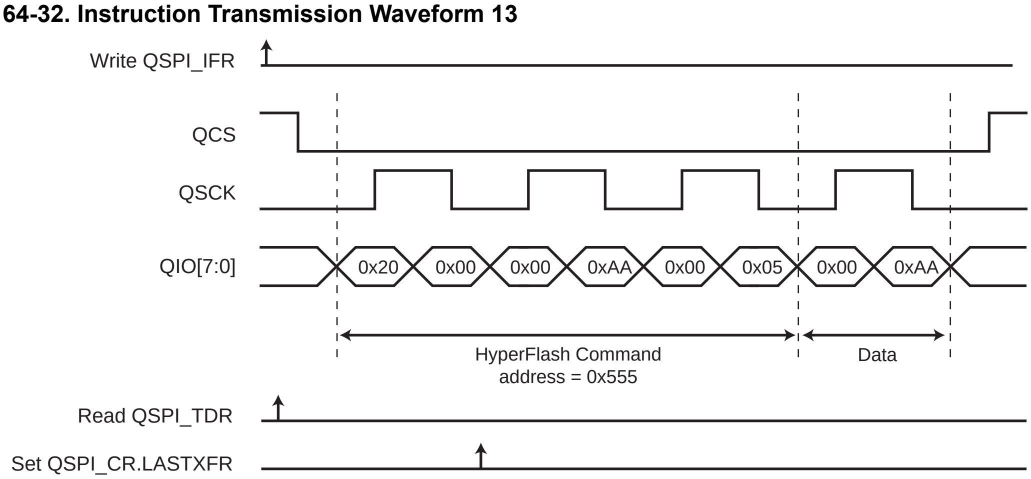
Example 13:

Classic HyperFlash command. Instruction in Octal DDR SPI, HyperFlash mode, without option, with data write, without dummy cycles, without fetch, write launched through APB interface.

Command: Data 0xAA at address 0x555

- Write 0x36C0_8899 in QSPI_IFR.
- Write 0x555 to QSPI_IADR.
- Update Configuration (See Updating the QSPI Configuration).
- Wait QSPI_ISR.TDRE flag.
- Write 0xAA to QSPI_TDR.
- Wait QSPI_ISR.TXEMPTY flag.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-32. Instruction Transmission Waveform 13



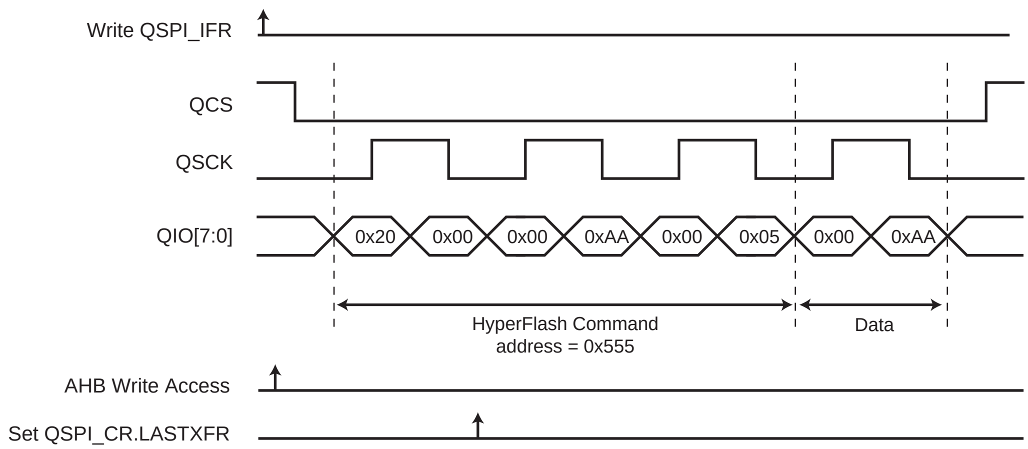
Example 14:

Classic HyperFlash command. Instruction in Octal DDR SPI, HyperFlash mode, without option, with data write, without dummy cycles, without fetch, write launched through the system bus interface.

Command: Data 0xAA at address 0x555

- Write 0x3640_8899 in QSPI_IFR.
- Write 0x01 in QSPI_WRACNT.
- Update Configuration (See Updating the QSPI Configuration).
- Write data 0xAA (byte access) in the QSPI system bus memory space with LSB address bytes = 0x555 (0x20000000–0x30000000).
- Wait for QSPI_ISR.LWRA to rise.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64-33. Instruction Transmission Waveform 14



64.6.10.9 Twin-Quad Mode

The Twin-Quad mode is activated by writing a '1' to QSPI_IFR.PROTTYP (Twin-Quad protocol). In this mode, the QSPI communicates with two Quad SPI memories as a single octal memory.

In this mode, only DUAL_CMD and OCT_CMD configurations are supported by QSPI_IFR.WIDTH.

When QSPI_IFR.TFRTYP is written to '1', programming is the same as in the standard Quad SPI protocol.

When QSPI_IFR.TFRTYP is written to '0', the programming differs due to the fact that each Quad SPI die has its own internal registers.

If QSPI_MR.SMRM is written to '0' (see Instruction Frame Transmission), a halfword read/write transfer must be issued instead of a byte transfer (one byte for each register of each die) to the Twin-Quad memory to write/read the registers of both die.

- For read transfers, the halfword read is as follows: {reg_byte[3:0], reg_byte[3:0], reg_byte[7:4], reg_byte[7:4]}.
  Example: If data 0xA7 is to be read in both registers of both die, the halfword read will be 0x77AA.
- For write transfers, the halfword must be sent as follows: {reg_byte[3:0], reg_byte[3:0], reg_byte[7:4], reg_byte[7:4]}.
  Example: If data 0xB5 is to be written in both registers of both die, the halfword to write is 0x55BB.

If QSPI_MR.SMRM is written to '1' (see Instruction Frame Transmission), a two-byte read/write transfer must be issued instead of a one-byte transfer (one byte for each register of each die) to the Twin-Quad memory to write/read the registers of both die.

- For read transfers, the first byte read is {reg_byte[3:0], reg_byte[3:0]}. The second byte read is {reg_byte[7:4], reg_byte[7:4]}.
  Example: If data 0xA7 is read in both registers of both die, the first byte read is 0xAA and the second byte read is 0x77.

- For write transfers, the first byte sent must be {reg_byte[3:0], reg_byte[3:0]}. The second byte sent must be {reg_byte[7:4], reg_byte[7:4]}.
  Example: If data 0xB5 is to be written in both registers of both die, the first byte to write is 0xBB and the second byte to write is 0x55.

If QSPI_IFR.WIDTH is configured to OCT_CMD, and DDREN is written to '1', then a halfword must be read/written instead of a byte for each QSPI_RDR or QSPI_TDR access.

In this configuration, the halfword to write in QSPI_TDR in case of a write transfer is {reg_byte[7:4], reg_byte[7:4], reg_byte[3:0], reg_byte[3:0]}.

Example: If data 0xB5 is to be written in both registers of both die, the halfword to write in QSPI_TDR is 0xBB55.

In this configuration, the halfword read in QSPI_RDR in case of a read transfer is {reg_byte[7:4], reg_byte[7:4], reg_byte[3:0], reg_byte[3:0]}.

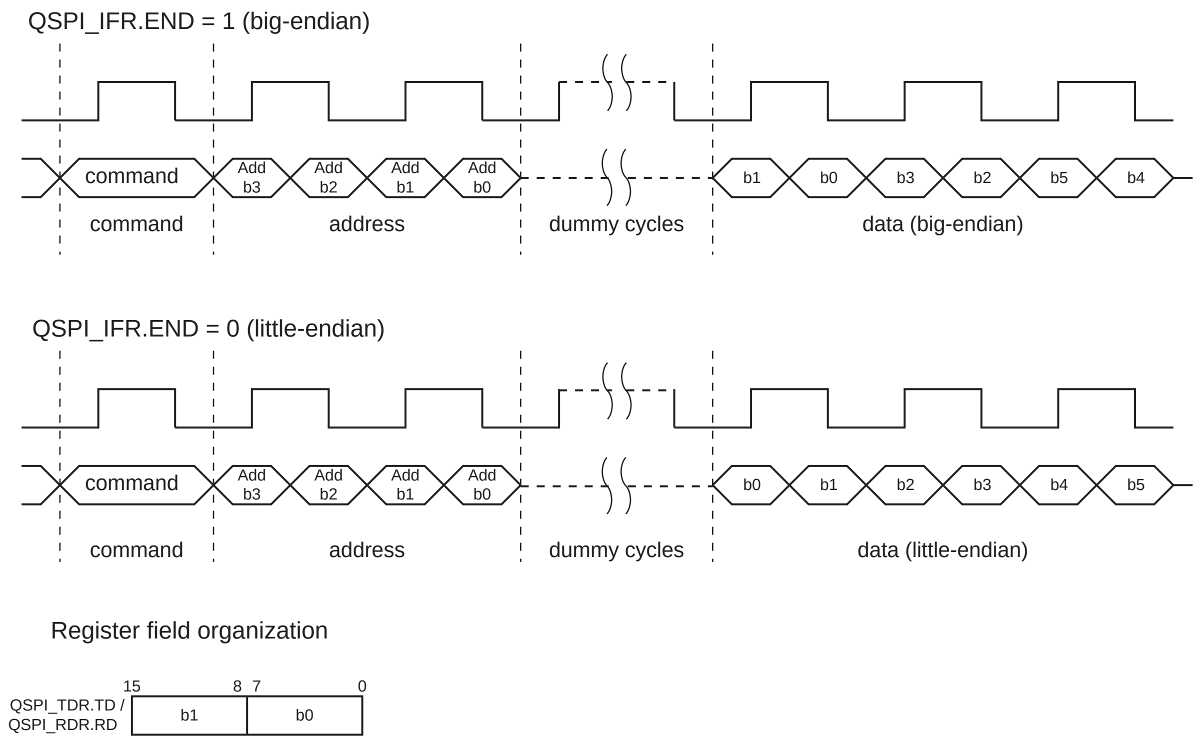
Example: If data 0xA7 is to be read in both registers of both die, the halfword read in QSPI_RDR is 0xAA77.

64.6.10.10 Endianness

If the QSPI is not configured in Octal DDR mode, endianness does not matter and QSPI_IFER.END should be left at '0' (default=little-endian).

When the QSPI is configured in Octal DDR mode, halfwords are sent/received instead of bytes. Endianness must be configured according to the memory datasheet.

Figure 64-34. Instruction Frame Endianness



64.6.10.11 Octal DDR Mode

Some memories support Octal DDR communication. To enable Octal DDR mode, configure QSPI_IFR.WIDTH to either '7', '8' or '9' and QSPI_IFR.DDREN to '1'. Configure the other parameters in QSPI_IFR to the targeted memory.

In this mode, QSPI_TDR.TD and QSPI_RDR.RD use the full 16-bit width.

Communication in Octal DDR mode is generally the same as for classic QSPI memories. However, there may be some differences when accessing memory registers. See Instruction Frame Transmission for details on accessing memory registers.

For memories using 8-bit registers:

- SMRM = 0
  The memory register can be read/written by performing a byte access in the QSPI memory space. In the case of a read, only the first byte read is considered. In the case of a write, check if the memory supports receiving the register value only on the first byte. If not, a halfword access can be performed to avoid confusion (refer to the memory datasheet to build the halfword).
- SMRM = 1
  In Octal DDR mode, QSPI_TDR.TD and QSPI_RDR.RD use the full 16-bit width. Therefore, it is mandatory to write/read a halfword in this configuration.

64.6.10.12 HyperFlash Mode

The QSPI supports HyperFlash memories. To enable HyperFlash mode, QSPI_IFR.PROTTYP must be written to ‘3’.

HyperFlash memories use Octal DDR communication. The recommendations given in Octal DDR Mode must be followed.

In HyperFlash memories, the address field is merged with the instruction field so QSPI_IFR.ADDREN must be set to ‘0’ in this mode even if the address is used. There is no use of instruction code, therefore QSPI_WICR and QSPI_RICR are not used in this mode.

Once HyperFlash mode is enabled, the procedure to access the memory is the same as for classic QSPI memories. See Instruction Frame Transmission.

For the HyperFlash Write Buffer procedure, QSPI_IFR.HFWBEN must be set. When this bit is set, a new command will be issued for each halfword written. In this mode, halfword size accesses are mandatory. see HyperFlash “Write Buffer” Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) and HyperFlash “Write Buffer” Flow Diagram TFRTYP = 1, Memory Write Access.

Note: In HyperFlash mode some bits of the HyperFlash command are automatically and seamlessly set. Thus the “Burst Type” bit of the HyperFlash command (bit 45) will always be set to 1 (linear Burst) using HyperFlash mode.

64.6.10.13 Time-out

The QSPI includes a time-out counter. This time-out counter detects any blocking state on the memory side (hardware connection issue, unknown command sent, etc,). If the QSPI is expecting data from the memory, it starts counting and if the counter reaches the value set in QSPI_TOUT.TCNTM, the flag QSPI_ISR.TOUT rises.

When the TOUT flag rises, any access waiting on the system bus interface is released with an error response so that the system bus is released. Any further access to the QSPI system bus interface receives an error response until the TOUT flag is cleared.

After a time-out error, it is mandatory to clear the TOUT flag by writing QSPI_CR.RTOUT to ‘1’. This also resets the QSPI internal state machines.

If QSPI_TOUT.TCNTM is set to ‘0’, the time-out feature is disabled.

64.6.11 Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data is scrambled when written to memory and unscrambled when data is read.

The external data lines can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the QSPI client device (e.g., memory).

The scrambling/unscrambling function can be enabled by writing a ‘1’ to the SCREN bit in the QSPI Scrambling Mode Register (QSPI_SMR).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable user scrambling key (field USRK) in the QSPI Scrambling Key Register (QSPI_SKR). QSPI_SKR is only accessible in Write mode.

When QSPI_SMR.SCRKL has been written once to ‘1’, QSPI_SKR.USRK cannot be written again until the next reset.

If QSPI_SMR.RVDIS is written to ‘0’, the scrambling/unscrambling algorithm includes the user scrambling key plus a random value depending on device processing characteristics. Data scrambled by a given microcontroller cannot be unscrambled by another.

If QSPI_SMR.RVDIS is written to ‘1’, the scrambling/unscrambling algorithm includes only the user scrambling key. No random value is part of the key.

The user scrambling key or the seed for key generation must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

64.6.11.1 Clearing Scrambling Keys on a Tamper Event

On a tamper detection event on WKUP0/1 pins (refer to the section "Shutdown Controller (SHDWC)"), an immediate clear of the scrambling key (QSPI_SKR) is performed if QSPI_MR.TAMPCLR is set.

64.6.12 Register Write Protection

To prevent any single software error from corrupting QSPI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the QSPI Write Protection Mode Register (QSPI_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the QSPI Write Protection Status Register (QSPI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the QSPI_WPSR.

The following registers can be write-protected when WPEN is set in QSPI_WPMR:

- QSPI Mode Register
- QSPI Serial Clock Register
- QSPI Instruction Address Register
- QSPI Write Instruction Code Register
- QSPI Instruction Frame Register
- QSPI Read Instruction Code Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register
- QSPI Refresh Register
- QSPI Write Access Counter Register
- QSPI DLL Configuration Register
- QSPI Pad Calibration Configuration Register
- QSPI Pad Calibration Bypass Register
- QSPI Timeout Register

The following registers can be write-protected when WPCREN is set in QSPI_WPMR:

- QSPI Control Register

The following registers can be write-protected when WPITEN is set in QSPI_WPMR:

- QSPI Interrupt Enable Register
- QSPI Interrupt Disable Register

64.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	QSPI_CR	31:24								LASTXFER
		23:16								
		15:8						RTOUT	STTFR	UPDCFG
		7:0	SWRST		SRFRSH	STPCAL	DLLOFF	DLLON	QSPIDIS	QSPIEN
0x04	QSPI_MR	31:24	DLYCS[7:0]							
		23:16	DLYBCT[7:0]							
		15:8	OENSD		QICMEN		NBBITS[3:0]			
		7:0	TAMPCLR		CSMODE[1:0]		DQSDLYEN	WDRBT		SMM
0x08	QSPI_RDR	31:24								
		23:16								
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0C	QSPI_TDR	31:24								
		23:16								
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x10	QSPI_ISR	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x14	QSPI_IER	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x18	QSPI_IDR	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x1C	QSPI_IMR	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x20	QSPI_SCR	31:24								
		23:16	DLYBS[7:0]							
		15:8								
		7:0							CPHA	CPOL
0x24	QSPI_SR	31:24								
		23:16								
		15:8								
		7:0		CALBSY	DLOCK	HIDLE	RBUSY	CSS	QSPIENS	SYNCBSY
0x28 ... 0x2F	Reserved									
0x30	QSPI_IAR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x34	QSPI_WICR	31:24								
		23:16	WROPT[7:0]							
		15:8	WRINST[15:8]							
		7:0	WRINST[7:0]							
0x38	QSPI_IFR	31:24			PROTTYP[1:0]		HFWBEN	DDRCMDEN	DQSEN	APBTFRTYP
		23:16	SMRM	END		NBDUM[4:0]				
		15:8	DDREN	CRM		TFRTYP	ADDRL[1:0]		OPTL[1:0]	
		7:0	DATAEN	OPTEN	ADDREN	INSTEN	WIDTH[3:0]			

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	QSPI_RICR	31:24								
		23:16	RDOPT[7:0]							
		15:8	RDINST[15:8]							
		7:0	RDINST[7:0]							
0x40	QSPI_SMR	31:24								
		23:16								
		15:8								
		7:0						SCRKL	RVDIS	SCREN
0x44	QSPI_SKR	31:24	USRK[31:24]							
		23:16	USRK[23:16]							
		15:8	USRK[15:8]							
		7:0	USRK[7:0]							
0x48 ... 0x4F	Reserved									
0x50	QSPI_REFRESH	31:24	REFRESH[31:24]							
		23:16	REFRESH[23:16]							
		15:8	REFRESH[15:8]							
		7:0	REFRESH[7:0]							
0x54	QSPI_WRACNT	31:24	NBWRA[31:24]							
		23:16	NBWRA[23:16]							
		15:8	NBWRA[15:8]							
		7:0	NBWRA[7:0]							
0x58	QSPI_DLLCFG	31:24								
		23:16								
		15:8								
		7:0								RANGE
0x5C	QSPI_PCALCFG	31:24	CALN[3:0]				CALP[3:0]			
		23:16								CALCNT[8]
		15:8	CALCNT[7:0]							
		7:0		CLKDIV[2:0]				DIFFPM	DAPCAL	AAON
0x60	QSPI_PCALBP	31:24								
		23:16					CALNBP[3:0]			
		15:8					CALPBP[3:0]			
		7:0								BPEN
0x64	QSPI_TOUT	31:24								
		23:16								
		15:8	TCNTM[15:8]							
		7:0	TCNTM[7:0]							
0x68 ... 0xE3	Reserved									
0xE4	QSPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	QSPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS

64.7.1 QSPI Control Register

Name: QSPI_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						RTOUT	STTFR	UPDCFG
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST		SRFRSH	STPCAL	DLLOFF	DLLON	QSPIDIS	QSPIEN
Access	W		W	W	W	W	W	W
Reset	–		–	–	–	–	–	–

Bit 24 – LASTXFER Last Transfer

Value	Description
0	No effect.
1	The chip select is deasserted after the end of character transmission.

Bit 10 – RTOUT Reset Time-out

Value	Description
0	No effect.
1	Request a TOUT flag reset.

Bit 9 – STTFR Start Transfer

Value	Description
0	No effect.
1	Starts the transfer when TFRTYP = 0 and SMRM = 1 or when DATAEN = 0.

Bit 8 – UPDCFG Update Configuration

Value	Description
0	No effect.
1	Requests an update of the QSPI Controller core configuration.

Bit 7 – SWRST QSPI Software Reset
DMA channels state, DLL state (see DLL) and QSPI_PADCTRL are not affected by a software reset.

Value	Description
0	No effect.
1	Resets the QSPI. A software-triggered hardware reset of the QSPI interface is performed.

Bit 5 – SRFRSH Start REFRESH

Value	Description
0	No effect.
1	Starts a refresh sequence. QSPI_ISR.RFRSHD indicates when the refresh sequence is over.

Bit 4 – STPCAL Start Pad Calibration

Value	Description
0	No effect.
1	Starts a QSPI pad calibration. QSPI_SR.CALBSY indicates the state of the calibration.

Bit 3 – DLLOFF DLL Off Request

Value	Description
0	No effect.
1	Disables the DLL. When the DLL is not locked (DLOCK flags in QSPI_SR), the QSPI Core does not receive a clock and is not functional.

Bit 2 – DLLON DLL On Request

Value	Description
0	No effect.
1	Enables the DLL. When the DLL is locked (DLOCK flags in QSPI_SR), the QSPI Core receives a clock and is functional.

Bit 1 – QSPIDIS QSPI Disable
As soon as QSPIDIS is set, the QSPI finishes its transfer.
All pins are set in Input mode and no data is received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disabled.
If QSPIEN and QSPIDIS are at ‘1’ when QSPI_CR is written, the QSPI is disabled.

Value	Description
0	No effect.
1	Disables the QSPI.

Bit 0 – QSPIEN QSPI Enable

Value	Description
0	No effect.
1	Enables the QSPI to transfer and receive data.

64.7.2 QSPI Mode Register

Name: QSPI_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OENSD		QICMEN		NBBITS[3:0]			
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAMPCLR		CSMODE[1:0]		DQSDLYEN	WDRBT		SMM
Access	R/W		R/W	R/W	R/W	R/W		R/W
Reset	0		0	0	0	0		0

Bits 31:24 – DLYCS[7:0] Minimum Inactive QCS Delay
Defines the minimum delay between the deactivation and the activation of QCS. The DLYCS time ensures the client minimum deselect time is respected.
If DLYCS written to ‘0’, one GCK period is inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × f_{GCK}

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
• SMM = 0
Defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
The following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × f_{GCK}) / 32
• SMM = 1
Defines the delay between last QSCK pulse and QCS rise.
When DLYBCT is written to ‘0’, no delay is inserted and the clock keeps its duty cycle over the character transfers.
The following equation determines the delay:
DLYBCT = Delay Between Consecutive Transfers × f_{GCK}

Bit 15 – OENSD Output Enable Shift Disabled

Value	Description
0	By default, the pad output enable signal is held for one QSCK period during the dummy cycles if QSPI.IFR_NBDUM is not equal to ‘0’.
1	The pad output enable signal is not held for one QSCK period during the dummy cycles.

Bit 13 – QICMEN QSPI Inter-chip Mode Enable

Value	Name	Description
0	DISABLED	QSPI_WICR.WROPT and QSPI_RICR.RDOPT define the transfer options and QSPI_IFR.NBDUM defines the number of dummy cycles.
1	ENABLED	No dummy cycles are inserted for write accesses, whatever the value configured in QSPI_IFR.NBDUM. The option field of the frame is automatically generated by the QSPI depending on the type of system bus transfer (the fields QSPI_RICR.RDOPT and QSPI_WICR.WROPT have no effect).

Bits 11:8 – NBBITS[3:0] Number Of Bits Per Transfer
NBBITS is used only when SMM is set to '0'.

Value	Name	Description
0	8_BIT	8 bits for transfer
8	16_BIT	16 bits for transfer

Bit 7 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on QSPI scrambling keys.
1	A tamper detection event immediately clears QSPI scrambling keys.

Bits 5:4 – CSMODE[1:0] Chip Select Mode
The CSMODE field determines how the chip select is deasserted.
This field is forced to LASTXFER when SMM is written to ‘1’.

Value	Name	Description
0	NOT_RELOADED	The chip select is deasserted if QSPI_TDR.TD has not been reloaded before the end of the current transfer.
1	LASTXFER	The chip select is deasserted when the bit LASTXFER is written to ‘1’ and the character written in QSPI_TDR.TD has been transferred.
2	SYSTEMATICALLY	The chip select is deasserted systematically after each transfer.

Bit 3 – DQSDLYEN DQS Delay Enable

Value	Name	Description
0	DISABLED	DQS Delay cell is disabled.
1	ENABLED	DQS Delay cell is enabled. The DQS Delay cell automatic refresh is triggered according to the configuration of QSPI_REFRESH.

Bit 2 – WDRBT Wait Data Read Before Transfer

Value	Name	Description
0	DISABLED	No effect. In SPI mode, a transfer can be initiated whatever the state of QSPI_RDR is.
1	ENABLED	In SPI mode, a transfer can start only if QSPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 0 – SMM Serial Memory Mode

Value	Name	Description
0	SPI	The QSPI is in SPI mode.
1	MEMORY	The QSPI is in Serial Memory mode.

64.7.3 QSPI Receive Data Register

Name: QSPI_RDR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RD[15:0] Receive Data
Data received by the QSPI is stored in this register right-justified. Unused bits read zero.
• QSPI_MR.SMM = 0
RD is defined by QSPI_MR.NBBITS.
• QSPI_MR.SMM = 1
RD is 8 bits or 16 bits in Octal DDR Mode.

64.7.4 QSPI Transmit Data Register

Name: QSPI_TDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the Transmit Data register in a right-justified format.
• QSPI_MR.SMM = 0
TD is defined by QSPI_MR.NBBITS field.
• QSPI_MR.SMM = 1
TD is 8 bits or 16 bits in Octal DDR Mode.

64.7.5 QSPI Interrupt Status Register

Name: QSPI_ISR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	RFRSHD
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 17 – TOUT QSPI Time-out

Value	Description
0	No QSPI time-out occurred since the last write of RTOUT bit on QSPI_CR.
1	At least one QSPI time-out occurred since the last write of RTOUT bit on QSPI_CR.

Bit 16 – RFRSHD Refresh Done

Value	Description
0	No refresh done event occurred since the last read of QSPI_ISR.
1	One refresh done event has been detected since the end of the last refresh command or the last read of QSPI_ISR.

Bit 15 – CSRA Chip Select Rise Autoclear

Value	Description
0	No chip select rise has been detected since beginning of the last command or the last read of QSPI_ISR.
1	One chip select rise has been detected since the beginning of the last command or the last read of QSPI_ISR.

Bit 14 – CSFA Chip Select Fall Autoclear

Value	Description
0	No chip select fall has been detected since end of the last command or the last read of QSPI_ISR.
1	One chip select fall has been detected since the end of the last command or the last read of QSPI_ISR.

Bit 13 – QITR QSPI Interrupt Rise

Value	Description
0	No rising of the QSPI memory interrupt line has been detected since the last read of QSPI_ISR.
1	At least one QSPI memory interrupt line rising edge occurred since the last read of QSPI_ISR.

Bit 12 – QITF QSPI Interrupt Fall

Value	Description
0	No falling of the QSPI memory interrupt line has been detected since the last read of QSPI_ISR.
1	At least one QSPI memory interrupt line falling edge occurred since the last read of QSPI_ISR.

Bit 11 – LWRA Last Write Access (cleared on read)

Value	Description
0	Last write access has not been sent since the last read of QSPI_SR or NBWRA = 0.
1	At least one last write access has been sent since the last read of QSPI_SR.

Bit 10 – INSTRE Instruction End Status (cleared on read)

Value	Description
0	No instruction end has been detected since the last read of QSPI_SR.
1	At least one instruction end has been detected since the last read of QSPI_SR.

Bit 8 – CSR Chip Select Rise (cleared on read)

Value	Description
0	No chip select rise has been detected since the last read of QSPI_SR.
1	At least one chip select rise has been detected since the last read of QSPI_SR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when QSPI_RDR is loaded at least twice from the serializer since the last read of the QSPI_RDR.

Value	Description
0	No overrun has been detected since the last read of QSPI_ISR.
1	At least one overrun error has occurred since the last read of QSPI_ISR.

Bit 2 – TXEMPTY Transmission Registers Empty (cleared by writing QSPI_TDR)

Value	Description
0	As soon as data is written in QSPI_TDR.
1	QSPI_TDR and the internal shifter are empty. If a transfer delay has been defined, TXEMPTY is set after the completion of such delay.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing QSPI_TDR)
TDRE equals zero when the QSPI is disabled or at reset. The QSPI enable command sets this bit to one.

Value	Description
0	Data has been written to QSPI_TDR and not yet transferred to the serializer.
1	The last data written in the QSPI_TDR has been transferred to the serializer.

Bit 0 – RDRF Receive Data Register Full (cleared by reading QSPI_RDR)

Value	Description
0	No data has been received since the last read of QSPI_RDR.
1	Data has been received and the received data has been transferred from the serializer to QSPI_RDR since the last read of QSPI_RDR.

64.7.6 QSPI Interrupt Enable Register

Name: QSPI_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	RFRSHD
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	W	W	W	W	W	W		W
Reset	–	–	–	–	–	–		–

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 17 – TOUT QSPI Time-out Interrupt Enable

Bit 16 – RFRSHD Refresh Done Interrupt Enable

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Enable

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Enable

Bit 13 – QITR QSPI Interrupt Rise Interrupt Enable

Bit 12 – QITF QSPI Interrupt Fall Interrupt Enable

Bit 11 – LWRA Last Write Access Interrupt Enable

Bit 10 – INSTRE Instruction End Interrupt Enable

Bit 8 – CSR Chip Select Rise Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – TXEMPTY Transmission Registers Empty Enable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

64.7.7 QSPI Interrupt Disable Register

Name: QSPI_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	RFRSHD
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	W	W	W	W	W	W		W
Reset	–	–	–	–	–	–		–

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 17 – TOUT QSPI Time-out Interrupt Disable

Bit 16 – RFRSHD Refresh Done Interrupt Disable

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Disable

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Disable

Bit 13 – QITR QSPI Interrupt Rise Interrupt Disable

Bit 12 – QITF QSPI Interrupt Fall Interrupt Disable

Bit 11 – LWRA Last Write Access Interrupt Disable

Bit 10 – INSTRE Instruction End Interrupt Disable

Bit 8 – CSR Chip Select Rise Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – TXEMPTY Transmission Registers Empty Disable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

64.7.8 QSPI Interrupt Mask Register

Name: QSPI_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	RFRSHD
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 17 – TOUT QSPI Time-out Interrupt Mask

Bit 16 – RFRSHD Refresh Done Interrupt Mask

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Mask

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Mask

Bit 13 – QITR QSPI Interrupt Rise Interrupt Mask

Bit 12 – QITF QSPI Interrupt Fall Interrupt Mask

Bit 11 – LWRA Last Write Access Interrupt Mask

Bit 10 – INSTRE Instruction End Interrupt Mask

Bit 8 – CSR Chip Select Rise Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – TXEMPTY Transmission Registers Empty Mask

Bit 1 – TDRE Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

64.7.9 QSPI Serial Clock Register

Name: QSPI_SCR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before QSCK
This field defines the delay from QCS valid to the first valid QSCK transition.
When DLYBS equals zero, the QCS valid to QSCK transition is 1/2 the QSCK clock period.
Otherwise, the following equation determines the delay:
DLYBS = Delay Before QSCK × f_{GCK}

Bit 1 – CPHA Clock Phase
CPHA determines which edge of QSCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data is captured on the leading edge of QSCK and changed on the following edge of QSCK.
1	Data is changed on the leading edge of QSCK and captured on the following edge of QSCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (QSCK). It is used with CPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of QSCK is logic level zero.
1	The inactive state value of QSCK is logic level one.

64.7.10 QSPI Status Register

Name: QSPI_SR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CALBSY	DLOCK	HIDLE	RBUSY	CSS	QSPIENS	SYNCBSY
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – CALBSY Pad Calibration Busy

Value	Description
0	Pad calibration is not ongoing.
1	Pad calibration is ongoing.

Bit 5 – DLOCK DLL Lock

Value	Description
0	DLL is not locked. The QSPI Controller and physical interface have not received a clock yet.
1	DLL is locked. The QSPI Controller and physical interface have received a clock.

Bit 4 – HIDLE QSPI Idle

Value	Description
0	The QSPI is not in Idle state (either transmitting or Chip Select is active).
1	The QSPI is in Idle state (not transmitting and Chip Select is inactive).

Bit 3 – RBUSY Read Busy

Value	Description
0	The system bus interface has no activity.
1	The system bus interface is currently processing accesses.

Bit 2 – CSS Chip Select Status

Value	Description
0	The chip select is asserted.
1	The chip select is not asserted.

Bit 1 – QSPIENS QSPI Enable Status

Value	Description
0	The QSPI is disabled.
1	The QSPI is enabled.

Bit 0 – SYNCBSY Synchronization Busy

Value	Description
0	No event synchronization between the QSPI Controller interface and QSPI Controller core is ongoing. Register accesses requiring synchronization are allowed, see Register Synchronization.
1	Event synchronization between the QSPI Controller interface and QSPI Controller core is ongoing. Register accesses requiring synchronization are not allowed, see Register Synchronization.

64.7.11 QSPI Instruction Address Register

Name: QSPI_IAR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Address
Address to send to the serial Flash memory in the instruction frame.

64.7.12 QSPI Write Instruction Code Register

Name: QSPI_WICR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	WROPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WRINST[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WRINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – WROPT[7:0] Write Option Code
Option code to send to the serial Flash memory in case of write transfer.

Bits 15:0 – WRINST[15:0] Write Instruction Code
Instruction code to send to the serial Flash memory in case of write transfer.

64.7.13 QSPI Instruction Frame Register

Name: QSPI_IFR
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			PROTTYP[1:0]		HFWBEN	DDRCMDEN	DQSEN	APBTFRTYP
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SMRM	END		NBDUM[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DDREN	CRM		TFRTYP	ADDRL[1:0]		OPTL[1:0]	
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATAEN	OPTEN	ADDREN	INSTEN	WIDTH[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:28 – PROTTYP[1:0] Protocol Type
Values not listed below are considered 'reserved'.

Value	Name	Description
0	STD_SPI	Standard (Q)SPI Protocol
1	TWIN_QUAD	Twin-Quad Protocol
2	OCTAFLASH	OctaFlash Protocol
3	HYPERFLASH	HyperFlash Protocol

Bit 27 – HFWBEN HyperFlash Write Buffer Enable

Value	Description
0	No effect.
1	Each write access received on the system bus interface generates a new command. IMPORTANT: Mandatory if HyperFlash Write Buffer feature is used.

Bit 26 – DDRCMDEN DDR Mode Command Enable

Value	Name	Description
0	DISABLED	Transfer of instruction field is performed in Single Data Rate mode even if DDREN is written to '1'.
1	ENABLED	Transfer of instruction field is performed in Double Data Rate mode if DDREN bit is written to '1'. If DDREN is written to '0', the instruction field is sent in Single Data Rate mode.

Bit 25 – DQSEN DQS Sampling Enable

Value	Description
0	Data from the memory are not sampled with DQS signal.
1	Data from the memory are sampled with DQS signal.

Bit 24 – APBTFRTYP Peripheral Bus Transfer Type

Value	Description
0	Peripheral bus register transfer to the memory is a write transfer. Used when TRFTYP is written to ‘0’ and SMRM to ‘1’.
1	Peripheral bus register transfer to the memory is a read transfer. Used when TRFTYP is written to ‘0’ and SMRM to ‘1’.

Bit 23 – SMRM Serial Memory Register Mode

Value	Description
0	Serial Memory registers are written via system bus access. See Instruction Frame Transmission for details.
1	Serial Memory registers are written via peripheral bus access. See Instruction Frame Transmission for details.

Bit 22 – END Endianness

Value	Description
0	Data are sent in little-endian format to the memory.
1	Data are sent in big-endian format to the memory.

Bits 20:16 – NBDUM[4:0] Number Of Dummy Cycles
Defines the number of dummy cycles required by the serial Flash memory before data transfer.

Bit 15 – DDREN DDR Mode Enable
DDRCMDEN defines how the instruction field is sent when Double Data Rate mode is enabled. If DDRCMDEN is at ‘0’, the instruction field is sent in Single Data Rate mode.

Value	Name	Description
0	DISABLED	Transfers are performed in Single Data Rate mode.
1	ENABLED	Transfers are performed in Double Data Rate mode, whereas the instruction field is still transferred in Single Data Rate mode.

Bit 14 – CRM Continuous Read Mode

Value	Name	Description
0	DISABLED	Continuous Read mode is disabled.
1	ENABLED	Continuous Read mode is enabled.

Bit 12 – TFRTYP Data Transfer Type

Value	Name	Description
0	TRSFR_REGISTER	Read/Write of memory register, write of memory page buffer. This configuration implies the following: • Either the AHB or the APB interface can be used to initiate the transfer (SMRM bit). • Returned data represents the memory register value. • If the APB interface is used, the RDRF and TDRE flags help to control the frame. • Scrambling is possible only if the APB interface is used. • For HyperFlash memories the “target" bit is set to register space in the HyperFlash header.

Value	Name	Description
1	TRSFR_MEMORY	Read/Write accesses to the memory space. This configuration implies the following: • Only the System Bus interface can be used to trigger accesses. • Access to random location is possible. • Address mask is applied and full size accesses only are performed. • Internal optimization algorithm is enabled to minimize latency. • Data are returned in a system bus protocol compliant way. • Seamless scrambling is enabled. • Seamless handling of HyperFlash Write Buffer programming command (one command for each data) • Seamless protocol-specific page boundary handling • Address shift is handled seamlessly (e.g. halfword memories).

Bits 11:10 – ADDRL[1:0] Address Length
The ADDRL bit determines the length of the address.

Value	Name	Description
0	8_BIT	8-bit address size
1	16_BIT	16-bit address size
2	24_BIT	24-bit address size
3	32_BIT	32-bit address size

Bits 9:8 – OPTL[1:0] Option Code Length
Determines the length of the option code. The value written in OPTL must be consistent with the value written in the field WIDTH. For example, OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).

Value	Name	Description
0	OPTION_1BIT	The option code is 1 bit long.
1	OPTION_2BIT	The option code is 2 bits long.
2	OPTION_4BIT	The option code is 4 bits long.
3	OPTION_8BIT	The option code is 8 bits long.

Bit 7 – DATAEN Data Enable

Value	Description
0	No data is sent/received to/from the serial Flash memory.
1	Data is sent/received to/from the serial Flash memory.

Bit 6 – OPTEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTEN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 3:0 – WIDTH[3:0] Width of Instruction Code, Address, Option Code and Data
OCT_OUTPUT, OCT_IO and OCT_CMD are supported only in DDR mode (QSPI_IFR.DDREN must be set). However, the instruction code can still be sent in SDR or DDR mode depending on DDRCMDEN configuration.

Value	Name	Description
0	SINGLE_BIT_SPI	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Single-bit SPI
1	DUAL_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Dual SPI
2	QUAD_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Quad SPI
3	DUAL_IO	Instruction: Single-bit SPI / Address-Option: Dual SPI / Data: Dual SPI
4	QUAD_IO	Instruction: Single-bit SPI / Address-Option: Quad SPI / Data: Quad SPI
5	DUAL_CMD	Instruction: Dual SPI / Address-Option: Dual SPI / Data: Dual SPI
6	QUAD_CMD	Instruction: Quad SPI / Address-Option: Quad SPI / Data: Quad SPI
7	OCT_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Octal SPI
8	OCT_IO	Instruction: Single-bit SPI / Address-Option: Octal SPI / Data: Octal SPI
9	OCT_CMD	Instruction: Octal SPI / Address-Option: Octal SPI / Data: Octal SPI

64.7.14 QSPI Read Instruction Code Register

Name: QSPI_RICR
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	RDOPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RDINST[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RDINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDOPT[7:0] Read Option Code
Option code to send to the serial Flash memory in case of read transfer.

Bits 15:0 – RDINST[15:0] Read Instruction Code
Instruction code to send to the serial Flash memory in case of read transfer.

64.7.15 QSPI Scrambling Mode Register

Name: QSPI_SMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SCRKL	RVDIS	SCREN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SCRKL Scrambling Key Lock

Value	Description
0	No action.
1	QSPI_SKR.USRK cannot be written until the next VDDCORE reset.

Bit 1 – RVDIS Scrambling/Unscrambling Random Value Disable

Value	Description
0	The scrambling/unscrambling algorithm includes the user scrambling key plus a random value that may differ between devices.
1	The scrambling/unscrambling algorithm includes only the user scrambling key.

Bit 0 – SCREN Scrambling/Unscrambling Enable

Value	Name	Description
0	DISABLED	The scrambling/unscrambling is disabled.
1	ENABLED	The scrambling/unscrambling is enabled.

64.7.16 QSPI Scrambling Key Register

Name: QSPI_SKR
Offset: 0x44
Reset: –
Property: Write-only

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USRK[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	USRK[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	USRK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
	USRK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – USRK[31:0] User Scrambling Key

64.7.17 QSPI Refresh Register

Name: QSPI_REFRESH
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	REFRESH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	REFRESH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	REFRESH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	REFRESH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – REFRESH[31:0] Refresh Delay Counter
Defines in GCK clock periods the delay between two refreshes of analog blocks.

64.7.18 QSPI Write Access Counter Register

Name: QSPI_WRACNT
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	NBWRA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NBWRA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NBWRA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NBWRA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NBWRA[31:0] Number of Write Accesses
Defines the number of write accesses before the rise of QSPI_ISR.LWRA.

64.7.19 QSPI DLL Configuration Register

Name: QSPI_DLLCFG
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

This register is not affected by a software reset (QSPI_CR.SWRST).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								RANGE
Access								R/W
Reset								0

Bit 0 – RANGE DLL Range

Value	Description
0	The QSPI Core clock is between 25 MHz and 100 MHz.
1	The QSPI Core clock is between 50 MHz and 208 MHz.

64.7.20 QSPI Pad Calibration Configuration Register

Name: QSPI_PCALCFG
Offset: 0x5C
Reset: 0x00000070
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

This register is not affected by a software reset (QSPI_CR.SWRST).

Bit	31	30	29	28	27	26	25	24
	CALN[3:0]				CALP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								CALCNT[8]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	CALCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CLKDIV[2:0]				DIFFPM	DAPCAL	AAON
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	1	1		0	0	0

Bits 31:28 – CALN[3:0] Calibration Code for N-channel

Bits 27:24 – CALP[3:0] Calibration Code for P-channel

Bits 16:8 – CALCNT[8:0] Pad Calibration Counter
Defines in Peripheral Clock periods the delay between the start of pad calibration analog circuitry and the calibration request.

Bits 6:4 – CLKDIV[2:0] Calibration Clock Division
The clock applied to the calibration cell is divided by CLKDIV + 1.

Bit 2 – DIFFPM Differential Pad Mode
This bit is not used by the QSPI core. Therefore it does not need to be synchronized using QSPI_CR.UPDCFG.

Value	Description
0	Pad differential mode is not enabled.
1	Pad differential mode is enabled.

Bit 1 – DAPCAL Disable Automatic Pad Calibration

Value	Description
0	Pad calibration is started automatically depending on the configuration of QSPI_REFRESH.
1	Pad calibration is not started automatically.

Bit 0 – AAON Analog Always On

Value	Description
0	The analog part of the pad calibration circuitry is switched off after each calibration (long delay for each calibration).

Value	Description
1	The analog part of the pad calibration circuitry is not switched off after each calibration (shorter delay after the first calibration).

64.7.21 QSPI Pad Calibration Bypass Register

Name: QSPI_PCALBP
Offset: 0x60
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

This register is not affected by a software reset (QSPI_CR.SWRST).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CALNBP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CALPBP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
								BPEN
Access								R/W
Reset								0

Bits 19:16 – CALNBP[3:0] Calibration Code Bypass for N-channel

Bits 11:8 – CALPBP[3:0] Calibration Code Bypass for P-channel

Bit 0 – BPEN Bypass Enable

Value	Description
0	Calibration code is not overridden by values of CALNBP and CALPBP.
1	Calibration code is overridden by values of CALNBP and CALPBP.

64.7.22 QSPI Timeout Register

Name: QSPI_TOUT
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TCNTM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCNTM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TCNTM[15:0] Time-out Counter Maximum Value
Indicates the time in GCK clock periods when the connected client does not answer and before the TOUT flag is set.

64.7.23 QSPI Write Protection Mode Register

Name: QSPI_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x515350	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register if WPKEY corresponds to 0x515350.
1	Enables the write protection on the Control register if WPKEY corresponds to 0x515350.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x515350 (“QSP” in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x515350 (“QSP” in ASCII)

64.7.24 QSPI Write Protection Status Register

Name: QSPI_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the QSPI_WPSR.
1	A write protection violation has occurred since the last read of the QSPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

65. Secure Digital MultiMedia Card Controller (SDMMC)

65.1 Description

The Secure Digital MultiMedia Card Controller (SDMMC) supports the embedded MultiMedia Card (e.MMC) Specification V5.1, the SD Memory Card Specification V3.0, and the SDIO V3.0 specification. It is compliant with the SD Host Controller Standard V3.0 specification.

The SDMMC includes the register set defined in the “SD Host Controller Simplified Specification V3.00” and additional registers to manage e.MMC devices, sampling tuning procedure, PAD calibration and enhanced features.

The SDMMC is clocked by two asynchronous clocks (HCLOCK, GCLK) (see Block Diagram) and requires the PMC to be configured first.

65.2 Embedded Characteristics

- Compatible with SD Host Controller Standard Specification Version 3.00
- Compatible with MultiMedia Card Specification Version V5.1
- Compatible with SD Memory Card Specification Version 3.00
- Compatible with SDIO Specification Version 3.00
- Support for 1-bit/4-bit SD/SDIO Devices
- Support for 1-bit/4-bit 8-bit e.MMC Devices
- Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
- Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)
- Support for SD/SDIO UHS-I SDR12 (Maximum SDCLK Frequency = 25 MHz)
- Support for SD/SDIO UHS-I SDR25 (Maximum SDCLK Frequency = 50 MHz)
- Support for SD/SDIO UHS-I SDR50 (Maximum SDCLK Frequency = 100 MHz)
- Support for SD/SDIO UHS-I SDR104 (Maximum SDCLK Frequency = 200 MHz)
- Support for SD/SDIO UHS-I DDR50 (Maximum SDCLK Frequency = 50 MHz)
- Support for SDSC, SDHC and SDXC
- Support for MMC/e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
- Support for MMC/e.MMC High Speed (Maximum SDCLK Frequency = 52 MHz)
- Support for e.MMC High Speed DDR (Maximum SDCLK Frequency = 52 MHz)
- Support for e.MMC HS200 (Maximum SDCLK Frequency = 200 MHz SDR)
- Support for e.MMC HS400 (Maximum SDCLK Frequency = 200 MHz DDR)
- e.MMC Boot Operation Mode Support
- Support for Block Size from 1 to 512 Bytes
- Support for Stream, Block and Multiblock Data Read and Write
  – Advanced DMA and SDMA capability
- Internal 1024-byte Dual Port RAM
- Support for both Synchronous and Asynchronous Abort
- Support for SDIO Card Interrupt

65.3 Embedded Features for SDMMC0/1/2

The device embeds three SDMMC interfaces:

- SDMMC0 supports all the features listed in Embedded Characteristics.
- SDMMC1 and SDMMC2 support the features listed in Embedded Characteristics except the following:
  – 8-bit e.MMC Devices

– e.MMC HS400

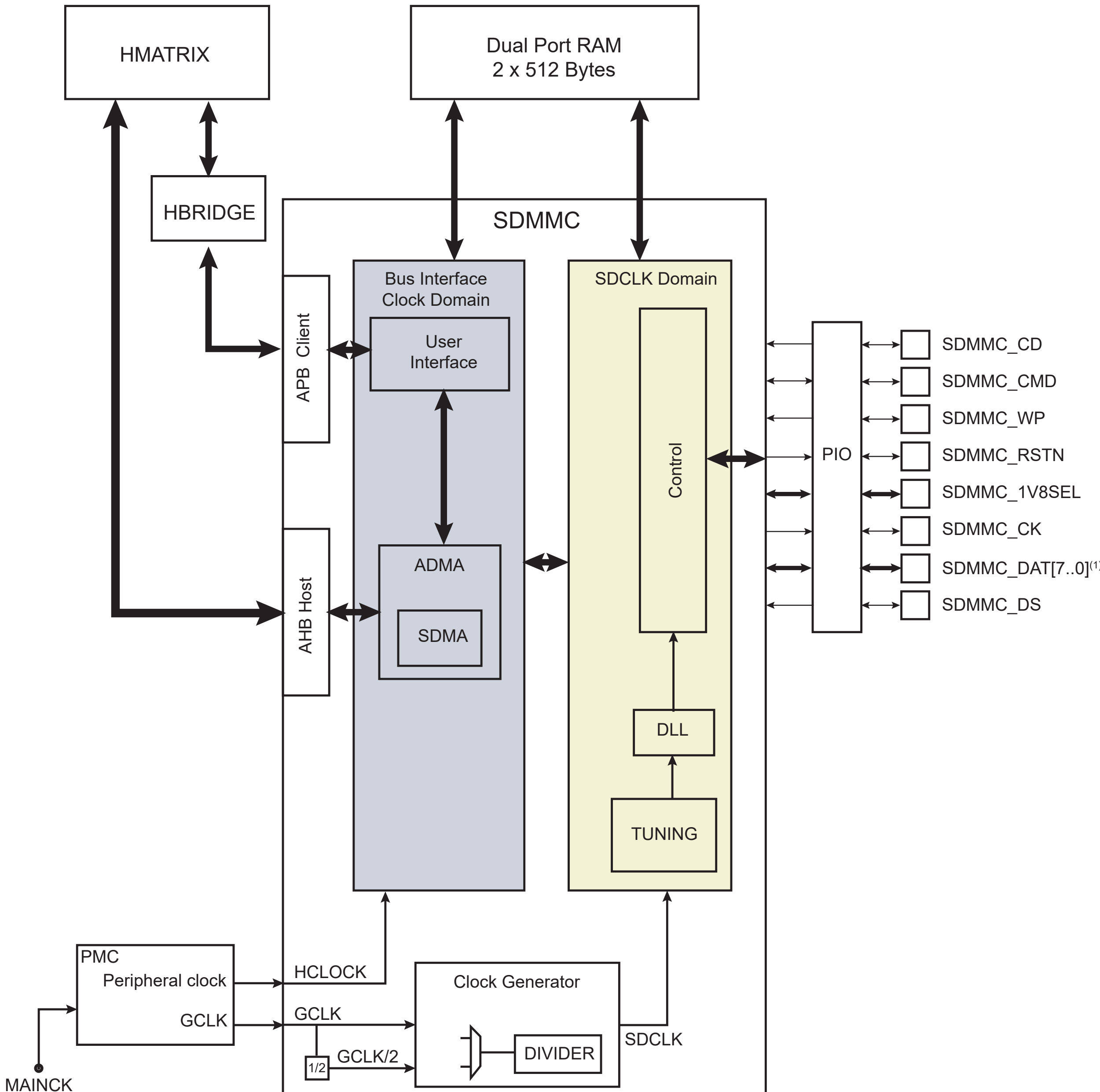
65.4 Reference Documents

Table 65-1. Reference Documents

Name	Link
SD Host Controller Simplified Specification V3.00	https://www.sdcard.org
SDIO Simplified Specification V3.00	
Physical Layer Simplified Specification V3.01	
Embedded MultiMedia Card (e.MMC) Electrical Standard 5.1	http://www.jedec.org

65.5 Block Diagram

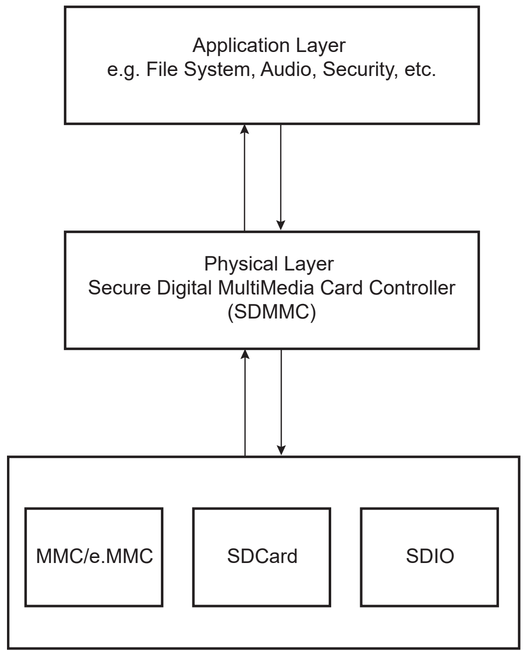
Figure 65-1. SDMMC Block Diagram



Note: 1. Limited to SDMMC_DAT[3..0] in SDMMC1 and SDMMC2.

65.6 Application Block Diagram

Figure 65-2. Application Block Diagram



65.7 Pin Name List

Table 65-2. I/O Lines Description for 8-bit Configuration

Pin Name	Pin Description	Type
SDMMC_CD	SDCard/SDIO/e.MMC Card Detect	Input
SDMMC_CMD	SDCard/SDIO/e.MMC Command/Response Line	I/O
SDMMC_WP	SDCard Connector Write Protect Signal	Input
SDMMC_RSTN	e.MMC Reset Signal	Output
SDMMC_1V8SEL	SDCard Signal Voltage Selection	Output
SDMMC_CK[(1)]	SDCard/SDIO/e.MMC Clock Signal	Output
SDMMC_DAT[7..0]	SDCard/SDIO/e.MMC Data Lines	I/O
SDMMC_DS	e.MMC HS400 Data Strobe	Input

Note:

1. When several SDMMCs are embedded in a product, SDMMC_CK refers to SDMMCx_CK, SDMMC_CMD to SDMMCx_CMD, SDMMC_DATy to SDMMCx_DATy, SDMMC_WP to SDMMCx_WP, SDMMC_1V8SEL to SDMMCx_1V8SEL, SDMMC_CD to SDMMCx_CD, SDMMC_DS to SDMMCx_DS and SDMMC_RSTN to SDMMCx_RSTN.

65.8 Product Dependencies

65.8.1 I/O Lines

The pins used for interfacing the Secure Digital MultiMedia Card (SDMMC) Controller are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the peripheral functions to SDMMC pins.

65.8.2 Power Management

The SDMMC is clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SDMMC clocks.

65.8.3 Interrupt Sources

The SDMMC has an interrupt line connected to the interrupt controller.

Handling the SDMMC interrupt requires programming the interrupt controller before configuring the SDMMC.

65.9 SD/SDIO Operating Mode

The SDMMC is fully compliant with the “SD Host Controller Simplified Specification V3.00” for SD/SDIO devices. See this specification for the SDMMC configuration.

See “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00” for SD/SDIO management.

65.10 e.MMC Operating Mode

The SDMMC supports management of e.MMC devices. As the “SD Host Controller Simplified Specification V3.00” does not apply to e.MMC devices, some registers have been added to those described in this specification in order to manage e.MMC devices. Most of the registers described in the “SD Host Controller Simplified Specification V3.00” must be used for e.MMC management, but e.MMC-specific features are managed using SDMMC_MC1R and SDMMC_MC2R.

65.10.1 Boot Operation Mode

In Boot Operation mode, the processor can read boot data from the e.MMC device by keeping the CMD line low after poweron before issuing the CMD1. The data can be read from either one of the boot partitions or the user area according to BOOT_PARTITION_ENABLE in the Extended CSD register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

65.10.1.1 Boot Procedure, Processor Mode

1. Configure the SDMMC:
   a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to the BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

   b. Select the speed mode (using SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD Register.

   c. Set the SDCLK frequency according to the selected speed mode.

   d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to ‘1’ (SDMMC_MC1R.BOOTA = 1).

   e. Enable the interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

   f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3)

   g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.DTDSEL = 1).

h. Select the NonDMA transfer (SDMMC_TMR.DMAEN = 0).

i. Optional: select the Auto CMD method (using SDMMC_TMR.ACMDEN).

j. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

k. Set the required number of read blocks (using SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to ‘1’.

2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).
4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

65.10.1.2 Boot Procedure, SDMA Mode

1. Configure SDMMC:
   a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

   b. Select the speed mode (SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD Register.

   c. Set the SDCLK frequency according to the selected speed mode.

   d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to 1 (SDMMC_MC1R.BOOTA = 1).

   e. Enable interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

   f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3).

   g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.TDSEL = 1).

   h. Select the SDMA transfer (SDMMC_TMR.DMAEN = 1 and SDMMC_HC1R.DMASEL = 0).

   i. Write the SDMA system address where the boot data will be copied (SDMMC_SSAR.ADDR).

   j. Optional: select the Auto CMD method (SDMMC_TMR.ACMDEN).
   Note: Auto CMD23 cannot be used with SDMA.

   k. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

   l. Set the required number of read blocks (SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to 1.
2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).
4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

65.10.1.3 Boot Procedure, ADMA Mode

1. Configure the SDMMC:
   a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

   b. Select the speed mode (SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD register.

   c. Set the SDCLK frequency according to the selected speed mode.

   d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to ‘1’ (SDMMC_MC1R.BOOTA = 1).

e. Enable interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3).

g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.DTDSEL = 1).

h. Select the ADMA transfer (SDMMC_TMR.DMAEN = 1 and SDMMC_HC1R.DMASEL = 2 or 3).

i. Write the address of the descriptor table in the ADMA system address (SDMMC_ASARx [0..1].ADMASA).

j. Optional: select the Auto CMD method (SDMMC_TMR.ACMDEN).

k. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

l. Set the required number of read blocks (SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to ‘1’.

2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation Mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).
4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

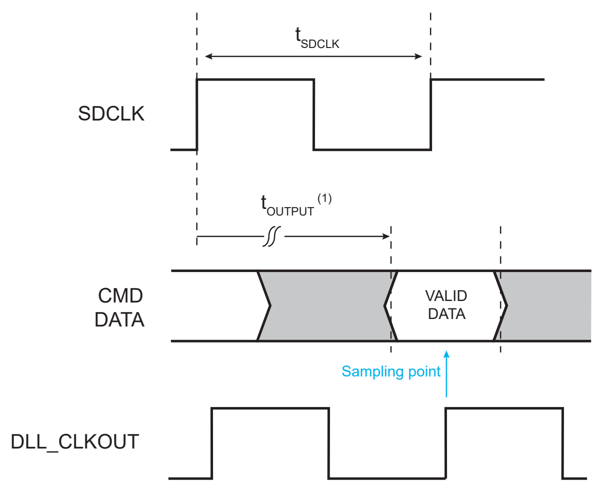
65.11 SDR104 / HS200 Tuning

65.11.1 DLL and Sampling Point

In SD/SDIO SDR104 mode (SDMMC_HC2R.VS18EN = 1 and SDMMC_HC2R.UHSMS = 3) or e.MMC HS200 mode (HS200EN = $B_{(hexa)}$), a tuning procedure must be performed first in order to adjust the sampling point for read transactions. For more details regarding the basic tuning procedure, see section “Sampling Clock Tuning Procedure” in the “SD Host Controller Simplified Specification V3.00” .

As the position of data and command coming from the device varies, a DLL is used to generate an accurate sampling point (DLL_CLKOUT) (see the figure below).

Figure 65-3. DLL Sampling Point

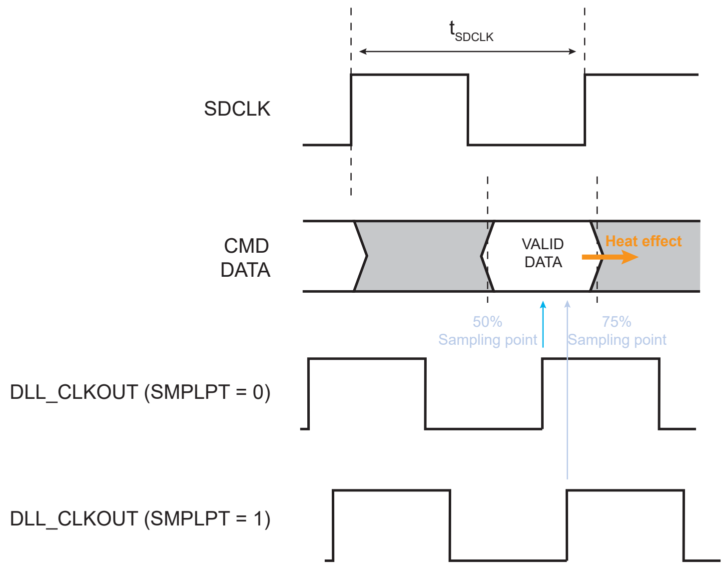


Note: 1. t_{OUTPUT} varies from 0 to 2 x t_{SDCLK}

The minimum SDLCK frequency is 100 MHz when SD/SDIO SDR104 or e.MMC HS200 is selected.

The sampling point can be selected to be located at 50% or 75% of the data window to anticipate the effect of the temperature rise. If SDMMC_TUNCR.SMPLPT is cleared, the sampling point is centered (50% of the data window). If SDMMC_TUNCR.SMPLPT is set to ‘1’, the sampling point is set at 75% of the data window (see the figure below).

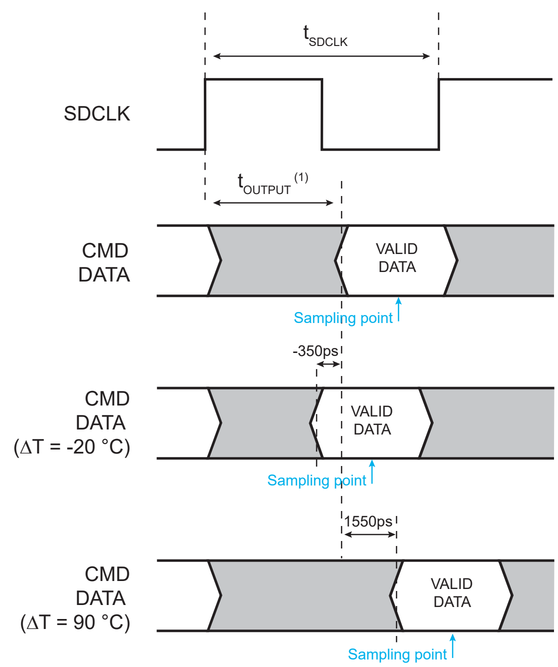
Figure 65-4. SDR104/HS200 Sampling Point Selection



65.11.2 Retuning Method

Once the data window sampling point has been tuned following the tuning procedure, the data window can be shifted by temperature drift. Thus, the tuning procedure must be applied periodically to adjust the sampling point position. The SDMMC implements a retuning timer which periodically instructs the software to restart the tuning procedure.

Figure 65-5. Temperature Effect on Data Window



Note: 1. t_{OUTPUT} varies from 0 to 2 x t_{SDCLK}

65.11.2.1 SDMMC Tuning Sequence

The SDMMC tuning sequence must only be done when SD/SDIO SDR104 , e.MMC HS200 or HS400 is selected and for a 100-MHz SDCLK frequency or higher.

1. Enable the retuning timer (SDMMC_RTC1R.TMREN = 1).

2. Configure the retuning period by setting SDMMC_RTCVR.TCVAL.
3. Set SDMMC_RTISTER.TEVT to ‘1’ so that the TEVT status flag in SDMMC_RTISTR rises each time the retuning timer counter period elapses.
4. Set SDMMC_RTISIER.TEVT to ‘1’ to generate an interrupt on the TEVT status flag assertion (optional).
5. Execute the tuning procedure as defined in “Sampling Clock Tuning Procedure” in the “SD Host Controller Simplified Specification V3.00” .
6. Start the retuning timer count (write SDMMC_RTC2R.RLD to 1). At this step, data can be read by the SDMMC.
7. Each time SDMMC_RTISTR.TEVT is set to ‘1’:
   a. Execute the tuning procedure as defined in “Sampling Clock Tuning Procedure” in the “SD Host Controller Simplified Specification V3.00” before issuing the next command.

   b. Restart the retuning timer count (write SDMMC_RTC2R.RLD to ‘1’).

   c. Resume data reading from the device.

When several instances of SDMMC are implemented in a product, the TEVT status flag of each SDMMC instance can be checked by reading SDMMC_RTSSR.

65.12 I/O Calibration

The need for output impedance calibration arises with higher data rates. As the data rate increases, some transmission line effects can occur and lead to the generation of undershoots and overshoots, hence degrading the signal quality.

To avoid these transmission problems, an I/O calibration cell is used to adjust the output impedance to the driven I/Os.

The I/O calibration sequence is mandatory when one of the SD/SDIO UHS-I modes (SDMMC_HC2R.VS18EN = 1) or e.MMC HS200 (HS200EN = $B_{(hexa)}$) is selected. It must be performed periodically to prevent the output impedance drift. Once the calibration is finished, the I/O calibration cell provides two four-bit control words (CALP[3:0] and CALN[3:0] in the Calibration Control register (SDMMC_CALCR)) to tune the output impedance, and thus reach the best transmission performances.

The I/O calibration sequence can be started manually by writing a ‘1’ to SDMMC_CALCR.EN. This bit is cleared automatically at the end of the calibration.

The I/O calibration sequence can also be performed automatically if SDMMC_CALCR.TUNDIS is cleared. In this case, the calibration starts automatically at the beginning of the tuning procedure when writing a ‘1’ to SDMMC_HC2R.EXTUN.

The I/O calibration cell requires a startup time defined by SDMMC_CALCR.CNTVAL. Thus, CNTVAL must be configured prior to start the calibration sequence. If SDMMC_CALCR.ALWYSON is set to ‘1’, the startup time is only required for the first calibration sequence as the analog circuitry is not shut down at the end of the calibration. In order to reduce the power consumption, the analog circuitry can be shut down at the end of the calibration sequence by clearing ALWYSON. In this case, the startup time is performed each time a calibration sequence is started.

65.13 e.MMC HS400 Timing Mode Selection

The HS400 mode has the following features:

- DDR data sampling method
- SDCLK frequency up to 200 MHz, Data rate is up to 400MB/s
- Only 8-bit bus width supported
- Signaling levels of 1.8V and 1.2V
- Data strobe signal is toggled only for data read, CRC response and CMD response (if Enhanced Strobe mode is supported)

HS400 mode is not supported during Boot Operation mode.

After the host driver initializes the device, it checks whether the device supports the HS400 mode by reading the field DEVICE_TYPE in the Extended CSD register of the e.MMC device (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 5.01”). Then it enables the HS400 mode in the device before changing the clock frequency to a frequency higher than 52 MHz.

65.13.1 Enhanced Strobe Mode Disabled

In order to switch to HS400 mode, the host driver must perform the following steps:

- Initialize the device.
- Select the device with CMD7.
- Read the DEVICE_TYPE[196] field of the Extended CSD register to validate whether the device supports HS400.
- Read the DRIVER_STRENGTH[197] field of the Extended CSD register to find the supported device drive strengths.
  Note: This step may be skipped if changes to driver strength are not needed.
- Set the “Selected Driver Strength” parameter in the HS_TIMING[185] field of the Extended CSD register to the appropriate driver strength for HS400 operation and set the “Timing Interface” parameter to 0x2 to switch the device to HS200 mode.
- Set SDMMC_HC2R.HS200EN to $B_{(HEXA)}$ to switch the controller to HS200 mode.
- Set the SDCLK frequency to the HS400 target operating frequency.
- Perform the tuning sequence (see section “SDMMC Tuning Sequence”). The tuning sequence in HS200 mode is required to synchronize the command response on the SDCMD line to SDCLK for HS400 operation.
- Set the “Timing Interface” parameter in the HS_TIMING[185] field of the Extended CSD register to 0x1 to switch the device to High Speed mode.
- Set SDMMC_HC1R.HSEN to ‘1’ to switch the controller to High Speed mode.
- Set the clock frequency to a value not greater than 52 MHz.
- Set BUS_WIDTH[183] to 0x06 to select the dual data rate x8 bus mode.
- Set the “Timing Interface” parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch the device to HS400 mode.
- Set SDMMC_MC3R.HS400EN to ‘1’ to switch the controller to HS400 mode.
- Change the SDCLK frequency to the HS400 operating frequency.

65.13.2 Enhanced Strobe Mode Enabled

In order to switch to HS400 mode, the host driver must perform the following steps:

- Initialize the device.
- Select the device with CMD7.
- Read the DEVICE_TYPE[196] field of the Extended CSD register to validate whether the device supports HS400.
- Read the STROBE_SUPPORT[184] field of the Extended CSD register to validate whether the device supports HS400.
- If supported, set the SDMMC_MC3R.ESMEN to ‘1’ to enable the Enhanced Strobe mode in the controller.
- Set the “Timing Interface” parameter in the HS_TIMING[185] field of the Extended CSD register to 0x1 to switch the device to High Speed mode.
- Set SDMMC_HC1R.HSEN to ‘1’ to switch the controller to High Speed mode.
- Set the clock frequency to a value not greater than 52 MHz.
- Read the DRIVER_STRENGTH[197] field of the Extended CSD register to find the supported device drive strengths.
  Note: This step may be skipped if changes to driver strength are not needed.
- Set the “Selected Driver Strength” parameter in the HS_TIMING[185] field of the Extended CSD register to the appropriate driver strength for HS400 operation.
- Set BUS_WIDTH[183] to 0x06 to select the dual data rate x8 bus mode.
- Set the “Timing Interface” parameter in the HS_TIMING [185] field of the Extended CSD register to 0x3 to switch the device to HS400 mode.

- Set SDMMC_MC3R.HS400EN to ‘1’ to switch controller to HS400 mode.
- Change the SDCLK frequency to the HS400 operating frequency.

65.14 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SDMMC_SSAR	31:24	ADDR/ARG2[31:24]							
		23:16	ADDR/ARG2[23:16]							
		15:8	ADDR/ARG2[15:8]							
		7:0	ADDR/ARG2[7:0]							
0x04	SDMMC_BSR	15:8		BOUNDARY[2:0]					BLKSIZE[9:8]	
		7:0	BLKSIZE[7:0]							
0x06	SDMMC_BCR	15:8	BLKCNT[15:8]							
		7:0	BLKCNT[7:0]							
0x08	SDMMC_ARG1R	31:24	ARG1[31:24]							
		23:16	ARG1[23:16]							
		15:8	ARG1[15:8]							
		7:0	ARG1[7:0]							
0x0C	SDMMC_TMR	15:8								
		7:0			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
0x0E	SDMMC_CR	15:8			CMDIDX[5:0]					
		7:0	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
0x10	SDMMC_RR0	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x14	SDMMC_RR1	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x18	SDMMC_RR2	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x1C	SDMMC_RR3	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x20	SDMMC_BDPR	31:24	BUFDATA[31:24]							
		23:16	BUFDATA[23:16]							
		15:8	BUFDATA[15:8]							
		7:0	BUFDATA[7:0]							
0x24	SDMMC_PSR	31:24								CMDLL
		23:16	DATLL[3:0]				WRPPL	CARDDPL	CARDSS	CARDINS
		15:8					BUFRDEN	BUFWREN	RTACT	WTACT
		7:0						DLACT	CMDINHD	CMDINHC
0x28	SDMMC_HC1R (SD_SDIO)	7:0	CARDDSEL	CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
0x28	SDMMC_HC1R (e.MMC)	7:0			EXTDW	DMASEL[1:0]		HSEN	DW	
0x29	SDMMC_PCR	7:0								SDBPWR
0x2A	SDMMC_BGCR (SD_SDIO)	7:0					INTBG	RWCTRL	CONTR	STPBGR
0x2A	SDMMC_BGCR (e.MMC)	7:0							CONTR	STPBGR
0x2B	SDMMC_WCR (SD_SDIO)	7:0						WKENCREM	WKENCINS	WKENCINT
0x2C	SDMMC_CCR	15:8	SDCLKFSEL[7:0]							
		7:0	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
0x2E	SDMMC_TCR	7:0					DTCVAL[3:0]			
0x2F	SDMMC_SRR	7:0						SWRSTDAT	SWRSTCMD	SWRSTALL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	SDMMC_NISTR (SD_SDIO)	15:8	ERRINT							CINT
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x30	SDMMC_NISTR (e.MMC)	15:8	ERRINT	BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x32	SDMMC_EISTR (SD_SDIO)	15:8						TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x32	SDMMC_EISTR (e.MMC)	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x34	SDMMC_NISTER (SD_SDIO)	15:8								CINT
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x34	SDMMC_NISTER (e.MMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x36	SDMMC_EISTER (SD_SDIO)	15:8						TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x36	SDMMC_EISTER (e.MMC)	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x38	SDMMC_NISIER (SD_SDIO)	15:8								CINT
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x38	SDMMC_NISIER (e.MMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x3A	SDMMC_EISIER (SD_SDIO)	15:8						TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3A	SDMMC_EISIER (e.MMC)	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3C	SDMMC_ACESR	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x3E	SDMMC_HC2R (SD_SDIO)	15:8	PVALEN	ASINTEN						
		7:0	SCLKSEL	EXTUN	DRVSEL[1:0]		VS18EN	UHSMS[2:0]		
0x3E	SDMMC_HC2R (e.MMC)	15:8	PVALEN							
		7:0	SCLKSEL	EXTUN	DRVSEL[1:0]		HS200EN[3:0]			
0x40	SDMMC_CA0R	31:24	SLTYPE[1:0]		ASINTSUP	SB64SUP		V18VSUP	V30VSUP	V33VSUP
		23:16	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP	MAXBLKL[1:0]	
		15:8	BASECLKF[7:0]							
		7:0	TEOCLKU		TEOCLKF[5:0]					
0x44	SDMMC_CA1R	31:24								
		23:16	CLKMULT[7:0]							
		15:8	RTMOD[1:0]		TSDR50		TCNTRT[3:0]			
		7:0		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
0x48	SDMMC_MCCAR	31:24								
		23:16	MAXCUR18V[7:0]							
		15:8	MAXCUR30V[7:0]							
		7:0	MAXCUR33V[7:0]							
0x4C ... 0x4F	Reserved									
0x50	SDMMC_FERACES	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x52	SDMMC_FEREIS	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x54	SDMMC_AESR	7:0						LMIS	ERRST[1:0]	
0x55 ... 0x57	Reserved									
0x58	SDMMC_ASAR0	31:24	ADMASA[31:24]							
		23:16	ADMASA[23:16]							
		15:8	ADMASA[15:8]							
		7:0	ADMASA[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x5C ... 0x5F	Reserved									
0x60	SDMMC_PVR0	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x62	SDMMC_PVR1	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x64	SDMMC_PVR2	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x66	SDMMC_PVR3	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x68	SDMMC_PVR4	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6A	SDMMC_PVR5	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6C	SDMMC_PVR6	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6E	SDMMC_PVR7	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x70 ... 0xFB	Reserved									
0xFC	SDMMC_SISR	15:8								
		7:0						INTSSL[2:0]		
0xFE	SDMMC_HCVR	15:8	VVER[7:0]							
		7:0	SVER[7:0]							
0x0100 ... 0x01FF	Reserved									
0x0200	SDMMC_APSR	31:24								
		23:16								
		15:8								
		7:0					HDATLL[3:0]			
0x0204	SDMMC_MC1R	7:0	FCD	RSTN	BOOTA	OPD	DDR		CMDTYP[1:0]	
0x0205	SDMMC_MC2R	7:0							ABOOT	SRESP
0x0206	SDMMC_MC3R	7:0			DQSUPVAL[2:0]				ESMEN	HS400EN
0x0207	SDMMC_DEBR	7:0							CDDVAL[1:0]	
0x0208	SDMMC_ACR	31:24								
		23:16								
		15:8	DFQOS[3:0]						BUFM[1:0]	
		7:0							BMAX[1:0]	
0x020C	SDMMC_CC2R	31:24								
		23:16								
		15:8								
		7:0								FSDCLKD
0x0210	SDMMC_RTC1R	7:0								TMREN
0x0211	SDMMC_RTC2R	7:0								RLD
0x0212 ... 0x0213	Reserved									
0x0214	SDMMC_RTCVR	31:24								
		23:16								
		15:8								
		7:0					TCVAL[3:0]			
0x0218	SDMMC_RTISTER	7:0								TEVT
0x0219	SDMMC_RTISIER	7:0								TEVT
0x021A ... 0x021B	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x021C	SDMMC_RTISTR	7:0								TEVT
0x021D	SDMMC_RTSSR	7:0						TEVTSLOT[2:0]		
0x021E ... 0x021F	Reserved									
0x0220	SDMMC_TUNCR	31:24								
		23:16								
		15:8								
		7:0								SMPLPT
0x0224 ... 0x022F	Reserved									
0x0230	SDMMC_CACR	31:24								
		23:16								
		15:8	KEY[7:0]							
		7:0								CAPWREN
0x0234	SDMMC_DBGR	31:24								
		23:16								
		15:8								
		7:0								NIDBG
0x0238 ... 0x023F	Reserved									
0x0240	SDMMC_CALCR	31:24	CALPBP[3:0]				CALP[3:0]			
		23:16	CALNBP[3:0]				CALN[3:0]			
		15:8	CNTVAL[7:0]							
		7:0		BPEN	TUNDIS	ALWYSON	CLKDIV[2:0]			EN
0x0244	SDMMC_EPVR8	15:8	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							

65.14.1 SDMMC SDMA System Address / Argument 2 Register

Name: SDMMC_SSAR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register contains the physical system memory address used for SDMA transfers or the second argument for Auto CMD23.

Bit	31	30	29	28	27	26	25	24
	ADDR/ARG2[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR/ARG2[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR/ARG2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR/ARG2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR/ARG2[31:0] SDMA System Address/Argument 2
ADDR: This field is the system memory address for a SDMA transfer. When the SDMMC stops an SDMA transfer, this field points to the system address of the next contiguous data position. This field can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. An interrupt can be generated to instruct the software to update this field. Writing the next system address of the next data position restarts the SDMA transfer.
ARG2: This field is used with Auto CMD23 to set a 32-bit block count value to the CMD23 argument while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by SDMMC_BCR. In this case, 65535 blocks is the maximum value.

65.14.2 SDMMC Block Size Register

Name: SDMMC_BSR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
		BOUNDARY[2:0]					BLKSIZE[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BLKSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 14:12 – BOUNDARY[2:0] SDMA Buffer Boundary
This field specifies the size of the contiguous buffer in the system memory. The SDMA transfer waits at every boundary specified by this field and the SDMMC generates the DMA Interrupt to instruct the software to update SDMMC_SSAR. If this field is set to 0 (buffer size = 4 Kbytes), the lowest 12 bits of SDMMC_SSAR.ADDRESS point to data in the contiguous buffer, and the upper 20 bits point to the location of the buffer in the system memory. This function is active when SDMMC_TMR.DMAEN is set.

Value	Name	Description
0	4K	4-Kbyte boundary
1	8K	8-Kbyte boundary
2	16K	16-Kbyte boundary
3	32K	32-Kbyte boundary
4	64K	64-Kbyte boundary
5	128K	128-Kbyte boundary
6	256k	256-Kbyte boundary
7	512K	512-Kbyte boundary

Bits 9:0 – BLKSIZE[9:0] Transfer Block Size
This field specifies the block size of data transfers for CMD14, CMD17, CMD18, CMD19, CMD24, CMD25, CMD53 and other data transfer commands such as CMD6, CMD8, ACMD13 and ACMD51. Values ranging from 1 to 512 can be set. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored.

65.14.3 SDMMC Block Count Register

Name: SDMMC_BCR
Offset: 0x06
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
	BLKCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BLKCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BLKCNT[15:0] Block Count for Current Transfer
This field is used only if SDMMC_TMR.BCEN (Block Count Enable) is set to 1 and is valid only for multiple block transfers. BLKCNT is the number of blocks to be transferred and it must be set to a value between 1 and the maximum block count. The SDMMC decrements the block count after each block transfer and stops when the count reaches 0. When this field is set to 0, no data block is transferred.
This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.
When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the previously saved block count is restored.

65.14.4 SDMMC Argument 1 Register

Name: SDMMC_ARG1R
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ARG1[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ARG1[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ARG1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ARG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ARG1[31:0] Argument 1
This register contains the SD command argument which is specified as the bit 39-8 of Command-Format in the "Physical Layer Simplified Specification V3.01" or "Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51" .

65.14.5 SDMMC Transfer Mode Register

Name: SDMMC_TMR
Offset: 0x0C
Reset: 0x0000
Property: Read/Write

This register is used to control data transfers. The user shall set this register before issuing a command which transfers data (see SDMMC_CR.DPSEL), or before issuing a Resume command. The user must save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, this register cannot be written while data transactions are in progress. Writes to this register are ignored when SDMMC_PSR.CMDINHD is 1.

Table 65-3. Determining the Transfer Type

MSBSEL	BCEN	BLKCNT (SDMMC_BCR)	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – MSBSEL Multi/Single Block Selection
This bit is set to 1 when issuing multiple-block transfer commands using DAT line(s). For any other commands, set this bit to 0. If this bit is 0, it is not necessary to set SDMMC_BCR (see the table Determing the Transfer Type).

Bit 4 – DTDSEL Data Transfer Direction Selection
This bit defines the direction of the DAT lines data transfers. Set this bit to 1 to transfer data from the device (SD Card/SDIO/e.MMC) to the SDMMC, and to 0 for all other commands.
0 (WRITE): Writes data from the SDMMC to the device.
1 (READ): Reads data from the device to the SDMMC.

Bits 3:2 – ACMDEN[1:0] Auto Command Enable
Two methods can be used to stop Multiple-block read and write operation:

- Auto CMD12: when the ACMDEN field is set to 1, the SDMMC issues CMD12 automatically when the last block transfer is completed. An Auto CMD12 error is indicated to SDMMC_ACESR. Auto CMD12 is not enabled if the command does not require CMD12.
- Auto CMD23: when the ACMDEN field is set to 2, the SDMMC issues a CMD23 automatically before issuing a command specified in SDMMC_CR.

The following conditions are required to use Auto CMD23:

- A memory card that supports CMD23 (SCR[33] = 1)
- If DMA is used, it must be ADMA (SDMA not supported).
- Only CMD18 or CMD25 is issued.

Note: The SDMMC does not check the command index.

Auto CMD23 can be used with or without ADMA. By writing SDMMC_CR, the SDMMC issues a CMD23 first and then issues a command specified by the SDMMC_CR.CMDIDX field. If CMD23 response errors are detected, the second command is not issued. A CMD23 error is indicated in SDMMC_ACESR. The CMD23 argument (32-bit block count value) is set in SDMMC_SSAR.
This field determines the use of auto command functions.

Value	Name	Description
0	DISABLED	Auto Command Disabled
1	CMD12	Auto CMD12 Enabled
2	CMD23	Auto CMD23 Enabled
3	–	Reserved

Bit 1 – BCEN Block Count Enable
This bit is used to enable SDMMC_BCR, which is only relevant for multiple block transfers. When this bit is 0, SDMMC_BCR is disabled, which is useful when executing an infinite transfer (see the table Determing the Transfer Type). If an ADMA2 transfer is more than 65535 blocks, this bit is set to 0 and the data transfer length is designated by the Descriptor Table.
0 (DISABLED): Block count is disabled.
1 (ENABLED): Block count is enabled.

Bit 0 – DMAEN DMA Enable
This bit enables the DMA functionality described in section “Supporting DMA” in “SD Host Controller Simplified Specification V3.00” . DMA can be enabled only if it is supported as indicated by the bit SDMMC_CA0R.ADMA2SUP. One of the DMA modes can be selected using the field SDMMC_HC1R.DMASEL. If DMA is not supported, this bit is meaningless and then always reads 0. When this bit is set to 1, a DMA operation begins when the user writes to the upper byte of SDMMC_CR.
0 (DISABLED): DMA functionality is disabled.
1 (ENABLED): DMA functionality is enabled.

65.14.6 SDMMC Command Register

Name: SDMMC_CR
Offset: 0x0E
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
			CMDIDX[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 13:8 – CMDIDX[5:0] Command Index
This bit shall be set to the command number (CMD0–63, ACMD0–63) that is specified in bits 45–40 of the Command-Format in the “Physical Layer Simplified Specification V3.01” , “SDIO Simplified Specification V3.00” , and “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .

Bits 7:6 – CMDTYP[1:0] Command Type

Value	Name	Description
0	NORMAL	Other commands
1	SUSPEND	CMD52 to write “Bus Suspend” in the Card Common Control registers (CCCR) (for SDIO only)
2	RESUME	CMD52 to write “Function Select” in the Card Common Control registers (CCCR) (for SDIO only)
3	ABORT	CMD12, CMD52 to write “I/O Abort” in the Card Common Control registers (CCCR) (for SDIO only)

Bit 5 – DPSEL Data Present Select
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT lines. It is set to 0 for the following:

- Commands using only CMD line (Ex. CMD52)
- Commands with no data transfer but using Busy signal on DAT[0] line (Ex. CMD38)
- Resume command

Value	Description
0	No data present
1	Data present

Bit 4 – CMDICEN Command Index Check Enable
If this bit is set to 1, the SDMMC checks the Index field in the response to see if it has the same value as the command index. If it has not, it is reported as a Command Index Error (CMDIDX) in SDMMC_EISTR. If this bit is set to 0, the Index field of the response is not checked.
0 (DISABLED): The Command Index Check is disabled.
1 (ENABLED): The Command Index Check is enabled.

Bit 3 – CMDCCEN Command CRC Check Enable
If this bit is set to 1, the SDMMC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error (CMDCRC) in SDMMC_EISTR. If this bit is set to 0, the CRC field is not checked. The position of the CRC field is determined according to the length of the response.
0 (DISABLED): The Command CRC Check is disabled.
1 (ENABLED): The Command CRC Check is enabled.

Bits 1:0 – RESPTYP[1:0] Response Type
This field is set according to the response type expected for the command index (CMDIDX).

Value	Name	Description
0	NORESP	No Response
1	RL136	Response Length 136
2	RL48	Response Length 48
3	RL48BUSY	Response Length 48 with Busy

65.14.7 SDMMC Response Register x

Name: SDMMC_RRx
Offset: 0x10 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CMDRESP[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CMDRESP[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMDRESP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMDRESP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CMDRESP[31:0] Command Response
The table below describes the mapping of command responses from the SD_SDIO/e.MMC bus to these registers for each responses type. In this table, R[] refers to a bit range of the response data as transmitted on the SD_SDIO/e.MMC bus.

Type of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R[39:8]	SDMMC_RR0[31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	SDMMC_RR3[31:0]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R[39:8]	SDMMC_RR3[31:0]
R2 (CID, CSD register)	CID or CSD register	R[127:8]	SDMMC_RR0[31:0] SDMMC_RR1[31:0] SDMMC_RR2[31:0] SDMMC_RR3[23:0]
R3 (OCR register)	OCR register for memory	R[39:8]	SDMMC_RR0[31:0]
R4 (OCR register)	OCR register for I/O	R[39:8]	SDMMC_RR0[31:0]
R5, R5b	SDIO response	R[39:8]	SDMMC_RR0[31:0]
R6 (Published RCA response)	New published RCA[31:16] and Card status bits	R[39:8]	SDMMC_RR0[31:0]

65.14.8 SDMMC Buffer Data Port Register

Name: SDMMC_BDPR
Offset: 0x20
Reset: –
Property: Read/Write

Note: The reset value is an unpredictable value read from the dual port RAM.

Bit	31	30	29	28	27	26	25	24
	BUFDATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	BUFDATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	BUFDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	BUFDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – BUFDATA[31:0] Buffer Data
The SDMMC data buffer can be accessed through this 32-bit Data Port register.

65.14.9 SDMMC Present State Register

Name: SDMMC_PSR
Offset: 0x24
Reset: 0x00F80000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								CMDLL
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	DATLL[3:0]				WRPPL	CARDDPL	CARDSS	CARDINS
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8
					BUFRDEN	BUFWREN	RTACT	WTACT
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DLACT	CMDINHD	CMDINHC
Access						R	R	R
Reset						0	0	0

Bit 24 – CMDLL CMD Line Level
This status is used to check the CMD line level to recover from errors, and for debugging.

Bits 23:20 – DATLL[3:0] DAT[3:0] Line Level
This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the Busy signal level from DAT[0].

Bit 19 – WRPPL Write Protect Pin Level
The Write Protect Switch is supported for memory and combo cards. This bit reflects the SDMMC_WP pin.

Value	Description
0	Write protected (SDMMC_WP = 0)
1	Write enabled (SDMMC_WP = 1)

Bit 18 – CARDDPL Card Detect Pin Level
This bit reflects the inverse value of the SDMMC_CD pin. Debouncing is not performed on this bit. This bit may be valid when CARDSS is set to 1, but it is not guaranteed because of the propagation delay. Use of this bit is limited to testing since it must be debounced by software.

Value	Description
0	No card present (SDMMC_CD = 1).
1	Card present (SDMMC_CD = 0).

Bit 17 – CARDSS Card State Stable
This bit is used for testing. If it is 0, the CARDDPL is not stable. If this bit is set to 1, it means that the CARDDPL is stable. No Card state can be detected if this bit is set to 1 and CARDINS is set to 0.
The Software Reset For All (SWRSTALL) in SDMMC_SRR does not affect this bit.

Value	Description
0	Reset or debouncing.
1	No card or card inserted.

Bit 16 – CARDINS Card Inserted
This bit indicates whether a card has been inserted. The SDMMC debounces this signal so that the user does not need to wait for it to stabilize.
A change from 0 to 1 raises the Card Insertion (CINS) status flag in SDMMC_NISTR if SDMMC_NISTER.CINS is set to 1. An interrupt is generated if SDMMC_NISIER.CINS is set to 1.
A change from 1 to 0 raises the Card Removal (CREM) status flag in SDMMC_NISTR if SDMMC_NISTER.CREM is set to 1. An interrupt is generated if SDMMC_NISIER.CREM is set to 1.
The Software Reset For All (SWRSTALL) in SDMMC_SRR does not affect this bit.

Bit 11 – BUFRDEN Buffer Read Enable
This bit is used for nonDMA read transfers. This flag indicates that valid data exists in the SDMMC data buffer. If this bit is 1, readable data exists in the buffer.
A change from 1 to 0 occurs when all the block data is read from the buffer.
A change from 0 to 1 occurs when block data is ready in the buffer. This raises the Buffer Read Ready (BRDRDY) status flag in SDMMC_NISTR if SDMMC_NISTER.BRDRDY is set to 1. An interrupt is generated if SDMMC_NISIER.BRDRDY is set to 1.

Bit 10 – BUFWREN Buffer Write Enable
This bit is used for nonDMA write transfers. This flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer.
A change from 1 to 0 occurs when all the block data are written to the buffer.
A change from 0 to 1 occurs when top of block data can be written to the buffer. This raises the Buffer Write Ready (BWRRDY) status flag in SDMMC_NISTR if SDMMC_NISTER.BWRRDY is set to 1. An interrupt is generated if SDMMC_NISIER.BWRRDY is set to 1.

Bit 9 – RTACT Read Transfer Active
This bit is used to detect completion of a read transfer. See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the read command.
- When a read operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- When the last data block as specified by Transfer Block Size (BLKSIZE) is transferred to the system.
- In case of ADMA2, end of read is designated by the descriptor table.
- When all valid data blocks in the SDMMC have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request (STPBGR) of SDMMC_BGCR being set to 1.

A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1.

Bit 8 – WTACT Write Transfer Active
This bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the SDMMC. See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the write command.
- When a write operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by the descriptor table.
- After getting the CRC status of any block where a data transmission is about to be stopped by a Stop At Block Gap Request (STPBGR) of SDMMC_BGCR.

During a write transaction and as the result of the Stop At Block Gap Request (STPBGR) being set, a change from 1 to 0 raises the Block Gap Event (BLKGE) status flag in SDMMC_NISTR if SDMMC_NISTER.BLKGE is set to 1. An interrupt is generated if BLKGE is set to 1 in SDMMC_NISIER. This status is useful to determine whether nonDAT line commands can be issued during Write Busy.

Bit 2 – DLACT DAT Line Active
This bit indicates whether one of the DAT lines on the bus is in use.
In the case of read transactions:

- This status indicates whether a read transfer is executing on the bus. A change from 1 to 0 resulting from setting the Stop At Block Gap Request (STPBGR) raises the Block Gap Event (BLKGE) status flag in SDMMC_NISTR if SDMMC_NISTER.BLKGE is set to 1. An interrupt is generated if SDMMC_NISIER.BLKGE is set to 1. See the section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
- This bit is set in either of the following cases:
  – After the end bit of the read command.
  – When writing 1 to SDMMC_BGCR.CONTR (Continue Request) to restart a read transfer.
- This bit is SDMMC cleared in either of the following cases:
  – When the end bit of the last data block is sent from the bus to the SDMMC. In case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
  – When a read transfer is stopped at the block gap initiated by a Stop At Block Gap Request (STPBGR).
- The SDMMC stops a read operation at the start of the interrupt cycle by driving the Read Wait (DAT[2] line) or by stopping the SD Clock. If the Read Wait signal is already driven (due to the fact that the data buffer cannot receive data), the SDMMC can continue to stop the read operation by driving the Read Wait signal. It is necessary to support the Read Wait in order to use the Suspend/Resume operation.

In the case of write transactions:

- This status indicates that a write transfer is executing on the bus. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. See the section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
- This bit is set in either of the following cases:
  – After the end bit of the write command.
  – When writing 1 to SDMMC_BGCR.CONTR (Continue Request) to continue a write transfer.
- This bit is cleared in either of the following cases:
  – When the card releases Write Busy of the last data block. If the card does not drive a Busy signal for 8 SDCLK, the SDMMC considers the card drive “Not Busy”. In the case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
  – When the card releases Write Busy prior to wait for write transfer as a result of a Stop At Block Gap Request (STPBGR).

Command with Busy:
This status indicates whether a command that indicates Busy (ex. erase command for memory) is executing on the bus. This bit is set to 1 after the end bit of the command with Busy and cleared when Busy is deasserted. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. See Figures 2.11 to 2.13 in the “SD Host Controller Simplified Specification V3.00” .

Value	Description
0	DAT line inactive.
1	DAT line active.

Bit 1 – CMDINHD Command Inhibit (DAT)
This status bit is 1 if either the DAT Line Active (DLACT) or the Read Transfer Active (RTACT) is set to 1. If this bit is 0, it indicates that the SDMMC can issue the next command. Commands with a Busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1.
Note: The software can save registers in the 000–00Dh range for a suspend transaction after this bit has changed from 1 to 0.

Value	Description
0	Can issue a command which uses the DAT line(s).
1	Cannot issue a command which uses the DAT line(s).

Bit 0 – CMDINHC Command Inhibit (CMD)

If this bit is 0, it indicates the CMD line is not in use and the SDMMC can issue a command using the CMD line. This bit is set to 1 immediately after SDMMC_CR is written. This bit is cleared when the command response is received. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the CMD12 or CMD23 response, but by the Read/Write command response.

Status issuing Auto CMD12 is not read from this bit. So, if a command is issued during Auto CMD12 operation, the SDMMC manages to issue both commands: CMD12 and a command set by SDMMC_CR.

Even if the Command Inhibit (DAT) is set to 1, commands using only the CMD line can be issued if this bit is 0.

A change from 1 to 0 raises the Command Complete (CMDC) status flag in SDMMC_NISTR if SDMMC_NISTER.CMDC is set to 1. An interrupt is generated if SDMMC_NISIER.CMDC is set to 1.

If the SDMMC cannot issue the command because of a command conflict error (see SDMMC_EISTR.CMDCRC) or because of a ‘Command Not Issued By Auto CMD12’ error (see section “SDMMC Auto CMD Error Status Register”), this bit remains 1 and Command Complete is not set.

Value	Description
0	Can issue a command using only CMD line.
1	Cannot issue a command.

65.14.10 SDMMC Host Control 1 Register (SD_SDIO)

Name: SDMMC_HC1R (SD_SDIO)
Offset: 0x28
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
	CARDDSEL	CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 7 – CARDDSEL Card Detect Signal Selection
This bit selects the source for the card detection.

Value	Description
0	The SDMMC_CD pin is selected.
1	The Card Detect Test Level (CARDDTL) is selected (for test purposes).

Bit 6 – CARDDTL Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection (CARDDSEL) is set to 1 and it indicates whether the card is inserted or not.

Value	Description
0	No card.
1	Card inserted.

Bits 4:3 – DMASEL[1:0] DMA Select
One of the supported DMA modes can be selected. The DMA modes supported are given in SDMMC_CA0R. Use of a selected DMA is determined by DMA Enable (DMAEN) in SDMMC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	–	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	–	Reserved

Bit 2 – HSEN High Speed Enable
Before setting this bit, the user must check High Speed Support (HSSUP) in SDMMC_CA0R.
If this bit is set to 0 (default), the SDMMC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, the user needs to reset SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.
Note: This bit is effective only if SDMMC_MC1R.DDR is set to 0.

Note: The clock divider (DIV) in the Clock Control register (SDMMC_CCR) must be set to a value different from 0 when HSEN is 1.

Value	Description
0	Normal Speed mode.
1	High Speed mode.

Bit 1 – DW Data Width
This bit selects the data width of the SDMMC. It must be set to match the data width of the card.
0 (1_BIT): 1-bit mode.
1 (4_BIT): 4-bit mode.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Bit 0 – LEDCTRL LED Control
This bit is used to caution the user not to remove the card while it is being accessed. If the software is going to issue multiple commands, this bit is set to 1 during all transactions.
0 (OFF): LED off.
1 (ON): LED on.

65.14.11 SDMMC Host Control 1 Register (e.MMC)

Name: SDMMC_HC1R (e.MMC)
Offset: 0x28
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	7	6	5	4	3	2	1	0
			EXTDW	DMASEL[1:0]		HSEN	DW	
Access			R/W	R/W	R/W	R/W	R/W	
Reset			0	0	0	0	0	

Bit 5 – EXTDW Extended Data Width
This bit controls the 8-bit Bus Width mode for embedded devices. Support of this function is indicated in 8-bit Support for Embedded Device in SDMMC_CA0R. If a device supports the 8-bit mode, this may be set to 1. If this bit is 0, the bus width is controlled by Data Width (DW).

Bits 4:3 – DMASEL[1:0] DMA Select
One of the supported DAM modes can be selected. The DMA modes supported are given in SDMMC_CA0R. Use of selected DMA is determined by DMA Enable (DMAEN) in SDMMC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	–	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	–	Reserved

Bit 2 – HSEN High Speed Enable
Before setting this bit, the user must check High Speed Support (HSSUP) in SDMMC_CA0R.
If this bit is set to 0 (default), the SDMMC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, the user needs to reset the SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.
Note: This bit is effective only if SDMMC_MC1R.DDR is set to 0.

Note: The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when HSEN is 1.

Value	Description
0	Normal Speed mode.
1	High Speed mode.

Bit 1 – DW Data Width
This bit selects the data width of the SDMMC. It must be set to match the data width of the card.
0 (1_BIT): 1-bit mode.
1 (4_BIT): 4-bit mode.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

65.14.12 SDMMC Power Control Register

Name: SDMMC_PCR
Offset: 0x29
Reset: 0x0E
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								SDBPWR
Access								R/W
Reset								0

Bit 0 – SDBPWR SD Bus Power
This bit is automatically cleared by the SDMMC if the card is removed. If this bit is cleared, the SDMMC stops driving SDMMC_CMD and SDMMC_DAT[7:0] (tri-state) and drives SDMMC_CK to low level.

65.14.13 SDMMC Block Gap Control Register (SD_SDIO)

Name: SDMMC_BGCR (SD_SDIO)
Offset: 0x2A
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
					INTBG	RWCTRL	CONTR	STPBGR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – INTBG Interrupt at Block Gap
This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the software detects an SDIO card insertion, it sets this bit according to the CCCR of the SDIO card.

Value	Name	Description
0	DISABLED	Interrupt detection disabled.
1	ENABLED	Interrupt detection enabled.

Bit 2 – RWCTRL Read Wait Control
The Read Wait control is optional for SDIO cards. If the card supports Read Wait, set this bit to enable use of the Read Wait protocol to stop read data using the SDMMC_DAT[2] line. Otherwise, the SDMMC stops the SDCLK to hold read data, which restricts command generation. When the software detects an SD card insertion, this bit must be set according to the CCCR of the SDIO card. If the card does not support Read Wait, this bit shall never be set to 1, otherwise an SDMMC_DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported.

Value	Description
0	Disables Read Wait control.
1	Enables Read Wait control.

Bit 1 – CONTR Continue Request
This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The SDMMC automatically clears this bit in either of the following cases:
– In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
– In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	No effect.
1	Restart.

Bit 0 – STPBGR Stop At Block Gap Request
This bit is used to stop executing read and write transactions at the next block gap for nonDMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in SDMMC_NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not.
During read transfers, the SDMMC stops the transaction by using the Read Wait signal (SDMMC_DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to SDMMC_BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to SDMMC_BDPR.

This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	Transfer
1	Stop

65.14.14 SDMMC Block Gap Control Register (e.MMC)

Name: SDMMC_BGCR (e.MMC)
Offset: 0x2A
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	7	6	5	4	3	2	1	0
							CONTR	STPBGR
Access							R/W	R/W
Reset							0	0

Bit 1 – CONTR Continue Request
This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The SDMMC automatically clears this bit in either of the following cases:
– In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
– In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	No effect.
1	Restart.

Bit 0 – STPBGR Stop At Block Gap Request
This bit is used to stop executing read and write transactions at the next block gap for nonDMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in SDMMC_NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not.
During read transfers, the SDMMC stops the transaction by using the Read Wait signal (SDMMC_DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to SDMMC_BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to SDMMC_BDPR.
This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	Transfer
1	Stop

65.14.15 SDMMC Wakeup Control Register (SD_SDIO)

Name: SDMMC_WCR (SD_SDIO)
Offset: 0x2B
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
						WKENCREM	WKENCINS	WKENCINT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WKENCREM Wakeup Event Enable on Card Removal
This bit enables a wakeup event via Card Removal (CREM) in SDMMC_NISTR. FN_WUS (Wakeup Support) in the CIS (Card Information Structure) does not affect this bit.
0 (DISABLED): Wakeup Event disabled.
1 (ENABLED): Wakeup Event enabled.

Bit 1 – WKENCINS Wakeup Event Enable on Card Insertion
This bit enables a wakeup event via Card Insertion (CINS) in SDMMC_NISTR. FN_WUS (Wakeup Support) in the CIS (Card Information Structure) does not affect this bit.
0 (DISABLED): Wakeup Event disabled.
1 (ENABLED): Wakeup Event enabled.

Bit 0 – WKENCINT Wakeup Event Enable on Card Interrupt
This bit enables a wakeup event via Card Interrupt (CINT) in SDMMC_NISTR. This bit can be set to 1 if FN_WUS (Wakeup Support) in the CIS (Card Information Structure) is set to 1 in the SDIO card.
0 (DISABLED): Wakeup Event disabled.
1 (ENABLED): Wakeup Event enabled.

65.14.16 SDMMC Clock Control Register

Name: SDMMC_CCR
Offset: 0x2C
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
	SDCLKFSEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 15:8 – SDCLKFSEL[7:0] SDCLK Frequency Select
This register is used to select the frequency of the SDCLK pin. There are two SDCLK Frequency modes according to Clock Generator Select (CLKGSEL).
The length of the clock divider (DIV) is extended to 10 bits (DIV[9:8] = USDCLKFSEL, DIV[7:0] = SDCLKFSEL)

- 10-bit Divided Clock Mode (CLKGSEL = 0): $f_{\mathrm{SDCLK}} = f_{\mathrm{BASECLK}}/(2 \times \mathrm{DIV})$. If DIV = 0 then $f_{\mathrm{SDCLK}} = f_{\mathrm{BASECLK}}$
- Programmable Clock Mode (CLKGSEL = 1): $f_{\mathrm{SDCLK}} = f_{\mathrm{MULTCLK}}/(\mathrm{DIV} + 1)$

This field depends on the setting of Preset Value Enable (PVALEN) in SDMMC_HC2R.
If PVALEN = 0, this field is set by the user.
If PVALEN = 1, this field is automatically set to a value specified in one of the SDMMC_PVR.

Bits 7:6 – USDCLKFSEL[1:0] Upper Bits of SDCLK Frequency Select
These bits expand the SDCLK Frequency Select (SDCLKFSEL) to 10 bits. These two bits are assigned to bit 09-08 of the clock divider as described in SDCLKFSEL.

Bit 5 – CLKGSEL Clock Generator Select
This bit is used to select the clock generator mode in the SDCLK Frequency Select field. If the Programmable mode is not supported (SDMMC_CA1R.CLKMULT (Clock Multiplier) set to 0), then this bit cannot be written and is always read at 0.
This bit depends on the setting of Preset Value Enable (PVALEN) in SDMMC_HC2R.
If PVALEN = 0, this bit is set by the user.
If PVALEN = 1, this bit is automatically set to a value specified in one of the SDMMC_PVRx.

Value	Description
0	Divided Clock mode (BASECLK is used to generate SDCLK).
1	Programmable Clock mode (MULTCLK is used to generate SDCLK).

Bit 2 – SDCLKEN SD Clock Enable
The SDMMC stops the SD Clock when writing this bit to 0. SDCLK Frequency Select (SDCLKFSEL) can be changed when this bit is 0. Then, the SDMMC maintains the same clock frequency until SDCLK is stopped (Stop at SDCLK = 0). If Card Inserted (CARDINS) in SDMMC_PSR is cleared, this bit is also cleared.

Value	Description
0	SD Clock disabled
1	SD Clock enabled

Bit 1 – INTCLKS Internal Clock Stable
This bit is set to 1 when the SD clock is stable after setting SDMMC_CCR.INTCLKEN (Internal Clock Enable) to 1. The user must wait to set SD Clock Enable (SDCLKEN) until this bit is set to 1.

Value	Description
0	Internal clock not ready.

Value	Description
1	Internal clock ready.

Bit 0 – INTCLKEN Internal Clock Enable
This bit is set to 0 when the SDMMC is not used or is awaiting a wakeup interrupt. In this case, its internal clock is stopped to reach a very low power state. Registers are still able to be read and written. The clock starts to oscillate when this bit is set to 1. Once the clock oscillation is stable, the SDMMC sets Internal Clock Stable (INTCLKS) in this register to 1.
This bit does not affect card detection.

Value	Description
0	The internal clock stops.
1	The internal clock oscillates.

65.14.17 SDMMC Timeout Control Register

Name: SDMMC_TCR
Offset: 0x2E
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
					DTCVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – DTCVAL[3:0] Data Timeout Counter Value
This value determines the interval at which DAT line timeouts are detected. For more information about timeout generation, see Data Timeout Error (DATTEO) in SDMMC_EISTR. When setting this register, the user can prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in SDMMC_EISTER).

$$\mathrm{TIMEOUT}_{(\mu s)} = \frac{2^{13\,+\,\mathrm{DTCVAL}}}{f_{\mathrm{FTEOCLK(MHz)}}}$$

Note: DTCVAL = $f_{(Hexa)}$ is reserved.

65.14.18 SDMMC Software Reset Register

Name: SDMMC_SRR
Offset: 0x2F
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
						SWRSTDAT	SWRSTCMD	SWRSTALL
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SWRSTDAT Software Reset for DAT Line
Only part of a data circuit is reset. The DMA circuit is also reset.
The following registers and bits are cleared by this bit:

- SDMMC Buffer Data Port Register
  – Buffer is cleared and initialized.
- SDMMC Present State Register
  – Buffer Read Enable (BUFRDEN)
  – Buffer Write Enable (BUFWREN)
  – Read Transfer Active (RTACT)
  – Write Transfer Active (WTACT)
  – DAT Line Active (DATLL)
  – Command Inhibit (DAT) (CMDINHD)
- SDMMC Block Gap Control Register (SD_SDIO)
  – Continue Request (CONTR)
  – Stop At Block Gap Request (STPBGR)
- SDMMC Normal Interrupt Status Register (SD_SDIO)
  – Buffer Read Ready (BRDRDY)
  – Buffer Write Ready (BWRRDY)
  – DMA Interrupt (DMAINT)
  – Block Gap Event (BLKGE)
  – Transfer Complete (TRFC)

Value	Description
0	Work
1	Reset

Bit 1 – SWRSTCMD Software Reset for CMD Line
Only part of a command circuit is reset.
The following registers and bits are cleared by this bit:

- SDMMC Present State Register
  – Command Inhibit (CMD) (CMDINHC)
- SDMMC Normal Interrupt Status Register (SD_SDIO) and SDMMC Normal Interrupt Status Register (e.MMC)
  – Command Complete (CMDC)

Value	Description
0	Work
1	Reset

Bit 0 – SWRSTALL Software Reset for All
This reset affects the entire SDMMC except the card detection circuit. During initialization, the SDMMC must be reset by setting this bit to '1'. This bit is automatically cleared to '0' when SDMMC_CA0R and SDMMC_CA1R are valid and the user can read them. If this bit is set to '1', the user should issue a reset command and reinitialize the card.

List of registers cleared to '0':

- SDMMC SDMA System Address / Argument 2 Register
- SDMMC Block Size Register
- SDMMC Block Count Register
- SDMMC Argument 1 Register
- SDMMC Command Register
- SDMMC Transfer Mode Register
- SDMMC Response Register
- SDMMC Buffer Data Port Register
- SDMMC Present State Register (except CMDLL, DATLL, WRPPL, CARDDDPL, CARDSS, CARDINS)
- SDMMC Host Control 1 Register (SD_SDIO)
- SDMMC Host Control 1 Register (e.MMC)
- SDMMC Power Control Register
- SDMMC Block Gap Control Register (SD_SDIO)
- SDMMC Block Gap Control Register (e.MMC)
- SDMMC Wakeup Control Register (SD_SDIO)
- SDMMC Clock Control Register
- SDMMC Timeout Control Register
- SDMMC Normal Interrupt Status Register (SD_SDIO)
- SDMMC Error Interrupt Status Register (SD_SDIO)
- SDMMC Normal Interrupt Status Enable Register (SD_SDIO)
- SDMMC Error Interrupt Status Enable Register (SD_SDIO)
- SDMMC Normal Interrupt Signal Enable Register (SD_SDIO)
- SDMMC Error Interrupt Signal Enable Register (SD_SDIO)
- SDMMC Auto CMD Error Status Register
- SDMMC Host Control 2 Register (SD_SDIO)
- SDMMC ADMA Error Status Register
- SDMMC ADMA System Address Register
- SDMMC Slot Interrupt Status Register
- SDMMC e.MMC Control 1 Register
- SDMMC e.MMC Control 2 Register
- SDMMC AHB Control Register
- SDMMC Clock Control 2 Register
- SDMMC Retuning Control 1 Register
- SDMMC Retuning Counter Value Register
- SDMMC Retuning Interrupt Status Enable Register
- SDMMC Retuning Interrupt Signal Enable Register
- SDMMC Retuning Interrupt Status Register
- SDMMC Tuning Control Register
- SDMMC Capabilities Control Register (except KEY)
- SDMMC Calibration Control Register (except CALN, CALP)

Value	Description
0	Work
1	Reset

65.14.19 SDMMC Normal Interrupt Status Register (SD_SDIO)

Name: SDMMC_NISTR (SD_SDIO)
Offset: 0x30
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
	ERRINT							CINT
Access	R							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – ERRINT Error Interrupt
If any of the bits in SDMMC_EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

Value	Description
0	No error.
1	Error.

Bit 8 – CINT Card Interrupt
Writing this bit to '1' does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the SDMMC detects the Card Interrupt without SDCLK to support wakeup. In 4-bit mode, the Card Interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the system.
When this bit is set to '1' and the user needs to start this interrupt service, Card Interrupt Status Enable (CINT) in SDMMC_NISTER may be set to '0' in order to clear the card interrupt statuses latched in the SDMMC and to stop driving the interrupt signal to the system. After completion of the card interrupt service (it should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set SDMMC_NISTER.CINT to '1' and start sampling the interrupt signal again.
Interrupt detected by DAT[1] is supported when there is one card per slot.
This bit can only be set to 1 if SDMMC_NISTER.CINT is set to 1. An interrupt can only be generated if SDMMC_NISIER.CINT is set to 1.

Value	Description
0	No card interrupt.
1	Card interrupt.

Bit 7 – CREM Card Removal
This status is set to '1' if Card Inserted (CARDINS) in SDMMC_PSR changes from '1' to '0'. When the user writes this bit to '1' to clear this status, the status of SDMMC_PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to '1' if SDMMC_NISTER.CREM is set to '1'. An interrupt can only be generated if SDMMC_NISIER.CREM is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Card state unstable or card inserted.
1	Card removed.

Bit 6 – CINS Card Insertion
This status is set if Card Inserted (CARDINS) in SDMMC_PSR changes from '0' to '1'. When the user writes this bit to '1' to clear this status, the status of SDMMC_PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to '1' if SDMMC_NISTER.CINS is set to '1'. An interrupt can only be generated if SDMMC_NISIER.CINS is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Card state unstable or card removed.
1	Card inserted.

Bit 5 – BRDRDY Buffer Read Ready
This status is set to '1' if the Buffer Read Enable (BUFRDEN) changes from '0' to '1'. See BUFRDEN in SDMMC_PSR. While processing the tuning procedure (Execute Tuning (EXTUN) in SDMMC_HC2R is set to '1'), BRDRDY is set to '1' for every CMD19 execution.
This bit can only be set to '1' if SDMMC_NISTER.BRDRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BRDRDY is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Not ready to read buffer.
1	Ready to read buffer.

Bit 4 – BWRRDY Buffer Write Ready
This status is set to '1' if the Buffer Write Enable (BUFWREN) changes from '0' to '1'. See BUFWREN in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.BWRRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BWRRDY is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Not ready to write buffer.
1	Ready to write buffer.

Bit 3 – DMAINT DMA Interrupt
This status is set if the SDMMC detects the Host SDMA Buffer boundary during transfer. See SDMA Buffer Boundary (BOUNDARY) in SDMMC_BSR.
In case of ADMA, by setting the “int” field in the descriptor table, the SDMMC raises this status flag when the descriptor line is completed. This status flag does not rise after Transfer Complete (TRFC).
This bit can only be set to '1' if SDMMC_NISTER.DMAINT is set to '1'. An interrupt can only be generated if SDMMC_NISIER.DMAINT is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No DMA Interrupt.
1	DMA Interrupt.

Bit 2 – BLKGE Block Gap Event
If the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR is set to 1, this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to 1, this bit is not set to 1.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit can only be set to '1' if SDMMC_NISTER.BLKGE is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BLKGE is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No block gap event.
1	Transaction stopped at block gap.

Bit 1 – TRFC Transfer Complete
This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data was read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR (after valid data was read to the system). See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in SDMMC_BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is deasserted. See DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.TRFC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.TRFC is set to '1'.
Writing this bit to '1' clears the bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to '1', execution of a command can be considered to be completed.

TRFC	DATTEO	Status Meaning
0	0	Interrupted by another factor
0	1	Timeout occurred during transfer
1	Don’t Care	Command execution complete

Value	Description
0	Command execution is not complete.
1	Command execution is complete.

Bit 0 – CMDC Command Complete
This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. See Command Inhibit (CMD) in SDMMC_PSR for details on how to control this bit.
This bit can only be set to 1 if SDMMC_NISTER.CMDC is set to 1. An interrupt can only be generated if SDMMC_NISIER.CMDC is set to 1.
Writing this bit to 1 clears the bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to 1, it can be considered that the response was not received correctly.

CMDC	CMDTEO	Status Meaning
0	0	Interrupted by another factor
Don’t care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Value	Description
0	No command complete.
1	Command complete.

65.14.20 SDMMC Normal Interrupt Status Register (e.MMC)

Name: SDMMC_NISTR (e.MMC)
Offset: 0x30
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
	ERRINT	BOOTAR						
Access	R	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 15 – ERRINT Error Interrupt
If any of the bits in SDMMC_EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

Value	Description
0	No error.
1	Error.

Bit 14 – BOOTAR Boot Acknowledge Received
This bit is set to '1' when the SDMMC received a Boot Acknowledge pattern from the e.MMC.
This bit can only be set to '1' if SDMMC_NISTER.BOOTAR is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BOOTAR is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Boot Acknowledge pattern not received.
1	Boot Acknowledge pattern received.

Bit 5 – BRDRDY Buffer Read Ready
This status is set to '1' if Buffer Read Enable (BUFRDEN) changes from '0' to '1'. See Buffer Read Enable (BUFRDEN) in SDMMC_PSR. While processing the tuning procedure (Execute Tuning (EXTUN) in SDMMC_HC2R is set to '1'), BRDRDY is set to '1' for every CMD19 execution.
This bit can only be set to '1' if SDMMC_NISTER.BRDRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BRDRDY is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Not ready to read buffer.
1	Ready to read buffer.

Bit 4 – BWRRDY Buffer Write Ready
This status is set to 1 if Buffer Write Enable (BUFWREN) changes from '0' to '1'. See Buffer Write Enable (BUFWREN) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.BWRRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BWRRDY is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	Not ready to write buffer.
1	Ready to write buffer.

Bit 3 – DMAINT DMA Interrupt
This status is set if the SDMMC detects the Host SDMA Buffer boundary during transfer. See SDMA Buffer Boundary (BOUNDARY) in SDMMC_BSR.
In case of ADMA, by setting "int" field in the descriptor table, the SDMMC raises this status flag when the descriptor line is completed. This status flag does not rise after the Transfer Complete (TRFC).
This bit can only be set to '1' if SDMMC_NISTER.DMAINT is set to '1'. An interrupt can only be generated if SDMMC_NISIER.DMAINT is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No DMA interrupt.
1	DMA interrupt.

Bit 2 – BLKGE Block Gap Event
If the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR is set to '1', this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to '1', this bit is not set to '1'.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. See section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). See section "Write Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
This bit can only be set to '1' if SDMMC_NISTER.BLKGE is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BLKGE is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No block gap event.
1	Transaction stopped at block gap.

Bit 1 – TRFC Transfer Complete
This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data was read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR (after valid data was read to the system). See section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in SDMMC_BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). See section "Write Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is deasserted. See DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.TRFC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.TRFC is set to '1'.
Writing this bit to '1' clears the bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to '1', execution of a command can be considered to be completed.

TRFC	DATTEO	Status Meaning
0	0	Interrupted by another factor
0	1	Timeout occurred during transfer

...........continued

TRFC	DATTEO	Status Meaning
1	Don’t Care	Command execution complete

Value	Description
0	Command execution is not complete.
1	Command execution is complete.

Bit 0 – CMDC Command Complete
This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. See CMRINHC in SDMMC_PSR for details on how to control this bit.
This bit can only be set to '1' if SDMMC_NISTER.CMDC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.CMDC is set to '1'.
Writing this bit to '1' clears the bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to '1', it can be considered that the response was not received correctly.

CMDC	CMDTEO	Status Meaning
0	0	Interrupted by another factor
Don’t care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Value	Description
0	No command complete.
1	Command complete.

65.14.21 SDMMC Error Interrupt Status Register (SD_SDIO)

Name: SDMMC_EISTR (SD_SDIO)
Offset: 0x32
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
						TUNING	ADMA	ACMD
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 10 – TUNING Tuning Error
This bit is set to 1 when an unrecoverable error is detected in a tuning circuit, except during a tuning procedure (occurrence of an error during a tuning procedure is indicated by Sampling Clock Select (SCLKSEL) in SDMMC_HC2R).
When detecting a tuning error, the user needs to abort the execution of a command and to perform tuning. To reset the tuning circuit, SCLKSEL must be set to 0 before executing the tuning procedure (see Figure 2-29 in the “SD Host Controller Simplified Specification V3.00”).
Tuning Error has a higher priority than the other error statuses generated during data transfer. When detecting a tuning error, the user should discard any data transferred by a current read/write command and retry the transfer after the SDMMC recovered from tuning circuit error.
This bit can only be set to '1' if SDMMC_EISTER.TUNING is set to '1'. An interrupt can only be generated if SDMMC_EISIER.TUNING is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 9 – ADMA ADMA Error
This bit is set to '1' when the SDMMC detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDMMC_AESR.
In addition, the SDMMC raises this status flag when it detects some invalid description data (Valid = 0) at the ST_FDS state (see section “Advanced DMA” in the “SD Host Controller Simplified Specification V3.00”). ADMA Error Status (ERRST) in SDMMC_AESR indicates that an error occurred in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to '1' if SDMMC_EISTER.ADMA is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ADMA is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 8 – ACMD Auto CMD Error
Auto CMD12 and Auto CMD23 use this error status. This bit is set to '1' when detecting that one of the 0 to 4 bits in SDMMC_ACESR[4:0] has changed from '0' to '1'. In the case of Auto CMD12, this bit is set to '1', not only when errors occur in Auto CMD12 but also when auto CMD12 is not executed due to the previous command error.
This bit can only be set to '1' if SDMMC_EISTER.ACMD is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ACMD is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 7 – CURLIM Current Limit Error
By setting SD Bus Power (SDBPWR) in SDMMC_PCR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0 means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.
This bit can only be set to '1' if SDMMC_EISTER.CURLIM is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 6 – DATEND Data End Bit Error
This bit is set to '1' either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to '1' if SDMMC_EISTER.DATEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 5 – DATCRC Data CRC error
This bit is set to '1' when detecting a CRC error when transferring read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.DATCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATCRC is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 4 – DATTEO Data Timeout Error
This bit is set to '1' when detecting one of following timeout conditions.
– Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SD Host Controller Simplified Specification V3.00”).
– Busy timeout after Write CRC status.
– Write CRC Status timeout.
– Read data timeout
This bit can only be set to '1' if SDMMC_EISTER.DATTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 3 – CMDIDX Command Index Error
This bit is set to '1' if a Command Index error occurs in the command response.
This bit can only be set to '1' if SDMMC_EISTER.CMDIDX is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.

Value	Description
1	Error.

Bit 2 – CMDEND Command End Bit Error
This bit is set to '1' when detecting that the end bit of a command response is 0.
This bit can only be set to '1' if SDMMC_EISTER.CMDEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDEND is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.
If a response is returned and the Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to '1' when detecting a CRC error in the command response.
The SDMMC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the SDMMC drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the SDMMC aborts the command (stops driving the CMD line) and sets this bit to '1'. CMDTEO is also set to '1' to indicate a CMD line conflict (see the table above).
This bit can only be set to '1' if SDMMC_EISTER.CMDCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDCRC is set to '1'.
Writing this bit to '1' clears the bit.

Bit 0 – CMDTEO Command Timeout Error
This bit is set to '1' only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the SDMMC detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to '1' as shown in the table below, this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the SDMMC.
This bit can only be set to '1' if SDMMC_EISTER.CMDTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDTEO is set to '1'.
Writing this bit to '1' clears the bit.

CMDCRC	CMDTEO	Error Types
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

65.14.22 SDMMC Error Interrupt Status Register (e.MMC)

Name: SDMMC_EISTR (e.MMC)
Offset: 0x32
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error
This bit is set to '1' when detecting that the e.MMC Boot Acknowledge Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.BOOTAE is set to '1'. An interrupt can only be generated if SDMMC_EISIER.BOOTAE is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 10 – TUNING Tuning Error
This bit is set to 1 when an unrecoverable error is detected in a tuning circuit, except during a tuning procedure (occurrence of an error during tuning procedure is indicated by Sampling Clock Select (SCLKSEL) in SDMMC_HC2R).
When detecting a tuning error, the user needs to abort the execution of a command and to perform tuning. To reset the tuning circuit, SCLKSEL must be set to 0 before executing the tuning procedure (see Figure 2-29 in the “SD Host Controller Simplified Specification V3.00”).
Tuning Error has a higher priority than the other error statuses generated during data transfer. When detecting a tuning error, the user should discard any data transferred by a current read/write command and retry the transfer after the SDMMC recovered from tuning circuit error.
This bit can only be set to '1' if SDMMC_EISTER.TUNING is set to '1'. An interrupt can only be generated if SDMMC_EISIER. TUNING is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 9 – ADMA ADMA Error
This bit is set to 1 when the SDMMC detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDMMC_AESR.
In addition, the SDMMC raises this status flag when it detects some invalid description data (Valid = 0) at the ST_FDS state (see section “Advanced DMA” in the “SD Host Controller Simplified Specification V3.00”). ADMA Error Status (ERRST) in SDMMC_AESR indicates that an error occurred in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to '1' if SDMMC_EISTER.ADMA is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ADMA is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.

Value	Description
1	Error.

Bit 8 – ACMD Auto CMD Error
Auto CMD12 and Auto CMD23 use this error status. This bit is set to '1' when detecting that one of the 0 to 4 bits in SDMMC_ACESR[4:0] has changed from 0 to 1. In the case of Auto CMD12, this bit is set to '1', not only when errors occur in Auto CMD12, but also when Auto CMD12 is not executed due to the previous command error.
This bit can only be set to '1' if SDMMC_EISTER.ACMD is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ACMD is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 7 – CURLIM Current Limit Error
By setting SD Bus Power (SDBPWR) in SDMMC_PSR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0 means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.
This bit can only be set to '1' if SDMMC_EISTER.CURLIM is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 6 – DATEND Data End Bit Error
This bit is set to '1' either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to '1' if SDMMC_EISTER.DATEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 5 – DATCRC Data CRC Error
This bit is set to '1' when detecting a CRC error during a transfer of read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.DATCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER. DATCRC is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 4 – DATTEO Data Timeout error
This bit is set to '1' when detecting one of following timeout conditions.
– Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
– Busy timeout after Write CRC Status.
– Write CRC Status timeout.
– Read data timeout
This bit can only be set to '1' if SDMMC_EISTER.DATTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 3 – CMDIDX Command Index Error
This bit is set to '1' if a Command Index error occurs in the command response.
This bit can only be set to '1' if SDMMC_EISTER.CMDIDX is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 2 – CMDEND Command End Bit Error
This bit is set to '1' when detecting that the end bit of a command response is 0.
This bit can only be set to '1' if SDMMC_EISTER.CMDEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDEND is set to '1'.
Writing this bit to '1' clears the bit.

Value	Description
0	No error.
1	Error.

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.
If a response is returned and Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to '1' when detecting a CRC error in the command response.
The SDMMC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the SDMMC drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the SDMMC aborts the command (stops driving the CMD line) and sets this bit to '1'. CMDTEO is also set to '1' to indicate a CMD line conflict (see he table “Relations between CMDCRC and CMDTEO”).
This bit can only be set to '1' if SDMMC_EISTER.CMDCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDCRC is set to '1'.
Writing this bit to 1 clears the bit.

Bit 0 – CMDTEO Command Timeout Error
This bit is set to '1' only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the SDMMC detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to '1' as shown in the table “Relations between CMDCRC and CMDTEO”, this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the SDMMC.
This bit can only be set to '1' if SDMMC_EISTER.CMDTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDTEO is set to '1'.
Writing this bit to '1' clears the bit.

CMDCRC	CMDTEO	Error Types
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

65.14.23 SDMMC Normal Interrupt Status Enable Register (SD_SDIO)

Name: SDMMC_NISTER (SD_SDIO)
Offset: 0x34
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
								CINT
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 8 – CINT Card Interrupt Status Enable
If this bit is set to 0, the SDMMC clears interrupt requests to the system. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The user may clear this bit before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.
0 (MASKED): The CINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CINT status flag in SDMMC_NISTR is enabled.

Bit 7 – CREM Card Removal Status Enable
0 (MASKED): The CREM status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CREM status flag in SDMMC_NISTR is enabled.

Bit 6 – CINS Card Insertion Status Enable
0 (MASKED): The CINS status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CINS status flag in SDMMC_NISTR is enabled.

Bit 5 – BRDRDY Buffer Read Ready Status Enable
0 (MASKED): The BRDRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BRDRDY status flag in SDMMC_NISTR is enabled.

Bit 4 – BWRRDY Buffer Write Ready Status Enable
0 (MASKED): The BWRRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BWRRDY status flag in SDMMC_NISTR is enabled.

Bit 3 – DMAINT DMA Interrupt Status Enable
0 (MASKED): The DMAINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The DMAINT status flag in SDMMC_NISTR is enabled.

Bit 2 – BLKGE Block Gap Event Status Enable
0 (MASKED): The BLKGE status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BLKGE status flag in SDMMC_NISTR is enabled.

Bit 1 – TRFC Transfer Complete Status Enable
0 (MASKED): The TRFC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The TRFC status flag in SDMMC_NISTR is enabled.

Bit 0 – CMDC Command Complete Status Enable
0 (MASKED): The CMDC status flag in SDMMC_NISTR is masked.

1 (ENABLED): The CMDC status flag in SDMMC_NISTR is enabled.

65.14.24 SDMMC Normal Interrupt Status Enable Register (e.MMC)

Name: SDMMC_NISTER (e.MMC)
Offset: 0x34
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
		BOOTAR						
Access		R/W						
Reset		0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Status Enable
0 (MASKED): The BOOTAR status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BOOTAR status flag in SDMMC_NISTR is enabled.

Bit 5 – BRDRDY Buffer Read Ready Status Enable
0 (MASKED): The BRDRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BRDRDY status flag in SDMMC_NISTR is enabled.

Bit 4 – BWRRDY Buffer Write Ready Status Enable
0 (MASKED): The BWRRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BWRRDY status flag in SDMMC_NISTR is enabled.

Bit 3 – DMAINT DMA Interrupt Status Enable
0 (MASKED): The DMAINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The DMAINT status flag in SDMMC_NISTR is enabled.

Bit 2 – BLKGE Block Gap Event Status Enable
0 (MASKED): The BLKGE status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BLKGE status flag in SDMMC_NISTR is enabled.

Bit 1 – TRFC Transfer Complete Status Enable
0 (MASKED): The TRFC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The TRFC status flag in SDMMC_NISTR is enabled.

Bit 0 – CMDC Command Complete Status Enable
0 (MASKED): The CMDC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CMDC status flag in SDMMC_NISTR is enabled.

65.14.25 SDMMC Error Interrupt Status Enable Register (SD_SDIO)

Name: SDMMC_EISTER (SD_SDIO)
Offset: 0x36
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
						TUNING	ADMA	ACMD
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 10 – TUNING Tuning Error Status Enable
0 (MASKED): The TUNING status flag in SDMMC_EISTR is masked.
1 (ENABLED): The TUNING status flag in SDMMC_EISTR is enabled.

Bit 9 – ADMA ADMA Error Status Enable
0 (MASKED): The ADMA status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ADMA status flag in SDMMC_EISTR is enabled.

Bit 8 – ACMD Auto CMD Error Status Enable
0 (MASKED): The ACMD status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ACMD status flag in SDMMC_EISTR is enabled.

Bit 7 – CURLIM Current Limit Error Status Enable
0 (MASKED): The CURLIM status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CURLIM status flag in SDMMC_EISTR is enabled.

Bit 6 – DATEND Data End Bit Error Status Enable
0 (MASKED): The DATEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATEND status flag in SDMMC_EISTR is enabled.

Bit 5 – DATCRC Data CRC Error Status Enable
0 (MASKED): The DATCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATCRC status flag in SDMMC_EISTR is enabled.

Bit 4 – DATTEO Data Timeout Error Status Enable
0 (MASKED): The DATTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATTEO status flag in SDMMC_EISTR is enabled.

Bit 3 – CMDIDX Command Index Error Status Enable
0 (MASKED): The CMDIDX status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDIDX status flag in SDMMC_EISTR is enabled.

Bit 2 – CMDEND Command End Bit Error Status Enable
0 (MASKED): The CMDEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDEND status flag in SDMMC_EISTR is enabled.

Bit 1 – CMDCRC Command CRC Error Status Enable
0 (MASKED): The CMDCRC status flag in SDMMC_EISTR is masked.

1 (ENABLED): The CMDCRC status flag in SDMMC_EISTR is enabled.

Bit 0 – CMDTEO Command Timeout Error Status Enable
0 (MASKED): The CMDTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDTEO status flag in SDMMC_EISTR is enabled.

65.14.26 SDMMC Error Interrupt Status Enable Register (e.MMC)

Name: SDMMC_EISTER (e.MMC)
Offset: 0x36
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Status Enable
0 (MASKED): The BOOTAE status flag in SDMMC_EISTR is masked.
1 (ENABLED): The BOOTAE status flag in SDMMC_EISTR is enabled.

Bit 10 – TUNING Tuning Error Status Enable
0 (MASKED): The TUNING status flag in SDMMC_EISTR is masked.
1 (ENABLED): The TUNING status flag in SDMMC_EISTR is enabled.

Bit 9 – ADMA ADMA Error Status Enable
0 (MASKED): The ADMA status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ADMA status flag in SDMMC_EISTR is enabled.

Bit 8 – ACMD Auto CMD Error Status Enable
0 (MASKED): The ACMD status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ACMD status flag in SDMMC_EISTR is enabled.

Bit 7 – CURLIM Current Limit Error Status Enable
0 (MASKED): The CURLIM status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CURLIM status flag in SDMMC_EISTR is enabled.

Bit 6 – DATEND Data End Bit Error Status Enable
0 (MASKED): The DATEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATEND status flag in SDMMC_EISTR is enabled.

Bit 5 – DATCRC Data CRC Error Status Enable
0 (MASKED): The DATCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATCRC status flag in SDMMC_EISTR is enabled.

Bit 4 – DATTEO Data Timeout Error Status Enable
0 (MASKED): The DATTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATTEO status flag in SDMMC_EISTR is enabled.

Bit 3 – CMDIDX Command Index Error Status Enable
0 (MASKED): The CMDIDX status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDIDX status flag in SDMMC_EISTR is enabled.

Bit 2 – CMDEND Command End Bit Error Status Enable
0 (MASKED): The CMDEND status flag in SDMMC_EISTR is masked.

1 (ENABLED): The CMDEND status flag in SDMMC_EISTR is enabled.

Bit 1 – CMDCRC Command CRC Error Status Enable
0 (MASKED): The CMDCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDCRC status flag in SDMMC_EISTR is enabled.

Bit 0 – CMDTEO Command Timeout Error Status Enable
0 (MASKED): The CMDTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDTEO status flag in SDMMC_EISTR is enabled.

65.14.27 SDMMC Normal Interrupt Signal Enable Register (SD_SDIO)

Name: SDMMC_NISIER (SD_SDIO)
Offset: 0x38
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
								CINT
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 8 – CINT Card Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the CINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CINT status rises in SDMMC_NISTR.

Bit 7 – CREM Card Removal Signal Enable
0 (MASKED): No interrupt is generated when the CREM status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CREM status rises in SDMMC_NISTR.

Bit 6 – CINS Card Insertion Signal Enable
0 (MASKED): No interrupt is generated when the CINS status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CINS status rises in SDMMC_NISTR.

Bit 5 – BRDRDY Buffer Read Ready Signal Enable
0 (MASKED): No interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.

Bit 4 – BWRRDY Buffer Write Ready Signal Enable
0 (MASKED): No interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.

Bit 3 – DMAINT DMA Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the DMAINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the DMAINT status rises in SDMMC_NISTR.

Bit 2 – BLKGE Block Gap Event Signal Enable
0 (MASKED): No interrupt is generated when the BLKGE status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BLKGE status rises in SDMMC_NISTR.

Bit 1 – TRFC Transfer Complete Signal Enable
0 (MASKED): No interrupt is generated when the TRFC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the TRFC status rises in SDMMC_NISTR.

Bit 0 – CMDC Command Complete Signal Enable
0 (MASKED): No interrupt is generated when the CMDC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CMDC status rises in SDMMC_NISTR.

65.14.28 SDMMC Normal Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_NISIER (e.MMC)
Offset: 0x38
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
		BOOTAR						
Access		R/W						
Reset		0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Signal Enable
0 (MASKED): No interrupt is generated when the BOOTAR status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BOOTAR status rises in SDMMC_NISTR.

Bit 5 – BRDRDY Buffer Read Ready Signal Enable
0 (MASKED): No interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.

Bit 4 – BWRRDY Buffer Write Ready Signal Enable
0 (MASKED): No interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.

Bit 3 – DMAINT DMA Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the DMAINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the DMAINT status rises in SDMMC_NISTR.

Bit 2 – BLKGE Block Gap Event Signal Enable
0 (MASKED): No interrupt is generated when the BLKGE status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BLKGE status rises in SDMMC_NISTR.

Bit 1 – TRFC Transfer Complete Signal Enable
0 (MASKED): No interrupt is generated when the TRFC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the TRFC status rises in SDMMC_NISTR.

Bit 0 – CMDC Command Complete Signal Enable
0 (MASKED): No interrupt is generated when the CMDC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CMDC status rises in SDMMC_NISTR.

65.14.29 SDMMC Error Interrupt Signal Enable Register (SD_SDIO)

Name: SDMMC_EISIER (SD_SDIO)
Offset: 0x3A
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
						TUNING	ADMA	ACMD
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 10 – TUNING Tuning Error Signal Enable
0 (MASKED): No interrupt is generated when the TUNING status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the TUNING status rises in SDMMC_EISTR.

Bit 9 – ADMA ADMA Error Signal Enable
0 (MASKED): No interrupt is generated when the ADMA status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ADMA status rises in SDMMC_EISTR.

Bit 8 – ACMD Auto CMD Error Signal Enable
0 (MASKED): No interrupt is generated when the ACMD status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ACMD status rises in SDMMC_EISTR.

Bit 7 – CURLIM Current Limit Error Signal Enable
0 (MASKED): No interrupt is generated when the CURLIM status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CURLIM status rises in SDMMC_EISTR.

Bit 6 – DATEND Data End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the DATEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATEND status rises in SDMMC_EISTR.

Bit 5 – DATCRC Data CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the DATCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATCRC status rises in SDMMC_EISTR.

Bit 4 – DATTEO Data Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the DATTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATTEO status rises in SDMMC_EISTR.

Bit 3 – CMDIDX Command Index Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.

Bit 2 – CMDEND Command End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDEND status rises in SDMMC_EISTR.

Bit 1 – CMDCRC Command CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the CDMCRC status rises in SDMMC_EISTR.

1 (ENABLED): An interrupt is generated when the CMDCRC status rises in SDMMC_EISTR.

Bit 0 – CMDTEO Command Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.

65.14.30 SDMMC Error Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_EISIER (e.MMC)
Offset: 0x3A
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Signal Enable
0 (MASKED): No interrupt is generated when the BOOTAE status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the BOOTAE status rises in SDMMC_EISTR.

Bit 10 – TUNING Tuning Error Signal Enable
0 (MASKED): No interrupt is generated when the TUNING status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the TUNING status rises in SDMMC_EISTR.

Bit 9 – ADMA ADMA Error Signal Enable
0 (MASKED): No interrupt is generated when the ADMA status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ADMA status rises in SDMMC_EISTR.

Bit 8 – ACMD Auto CMD Error Signal Enable
0 (MASKED): No interrupt is generated when the ACMD status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ACMD status rises in SDMMC_EISTR.

Bit 7 – CURLIM Current Limit Error Signal Enable
0 (MASKED): No interrupt is generated when the CURLIM status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CURLIM status rises in SDMMC_EISTR.

Bit 6 – DATEND Data End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the DATEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATEND status rises in SDMMC_EISTR.

Bit 5 – DATCRC Data CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the DATCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATCRC status rises in SDMMC_EISTR.

Bit 4 – DATTEO Data Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the DATTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATTEO status rises in SDMMC_EISTR.

Bit 3 – CMDIDX Command Index Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.

Bit 2 – CMDEND Command End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDEND status rises in SDMMC_EISTR.

1 (ENABLED): An interrupt is generated when the CMDEND status rises in SDMMC_EISTR.

Bit 1 – CMDCRC Command CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the CDMCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDCRC status rises in SDMMC_EISTR.

Bit 0 – CMDTEO Command Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.

65.14.31 SDMMC Auto CMD Error Status Register

Name: SDMMC_ACESR
Offset: 0x3C
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – CMDNI Command Not Issued by Auto CMD12 Error
This bit is set to 1 when CMD_wo_DAT is not executed due to an Auto CMD12 error (SDMMC_ACESR[4:1]). This bit is set to 0 when Auto CMD Error is generated by Auto CMD23.

Value	Description
0	No error.
1	Error.

Bit 4 – ACMDIDX Auto CMD Index Error
This bit is set to 1 when the Command Index error occurs in response to a command.

Value	Description
0	No error.
1	Error.

Bit 3 – ACMDEND Auto CMD End Bit Error
This bit is set to 1 when detecting that the end bit of the command response is 0.

Value	Description
0	No error.
1	Error.

Bit 2 – ACMDCRC Auto CMD CRC Error
This bit is set to 1 when detecting a CRC error in the command response (see table Relation between ACMDCRC and ACMDTEO for more details).

Bit 1 – ACMDTEO Auto CMD Timeout Error
This bit is set to 1 if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (SDMMC_ACESR[4:2]) are meaningless.

Table 65-4. Relation between ACMDCRC and ACMDTEO

ACMDCRC	ACMDTEO	Error Types
0	0	No error
0	1	Response Timeout error
1	0	Response CRC error
1	1	CMD line conflict

Bit 0 – ACMD12NE Auto CMD12 Not Executed
If a memory multiple block data transfer is not started due to a command error, this bit is not set to 1 because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the SDMMC cannot issue Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (SDMMC_ACESR[4:1]) are meaningless.

This bit is set to 0 when an Auto CMD error is generated by Auto CMD23.

Value	Description
0	No error.
1	Error.

65.14.32 SDMMC Host Control 2 Register (SD_SDIO)

Name: SDMMC_HC2R (SD_SDIO)
Offset: 0x3E
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
	PVALEN	ASINTEN						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
	SCLKSEL	EXTUN	DRVSEL[1:0]		VS18EN	UHSMS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – PVALEN Preset Value Enable
As the operating SDCLK frequency and I/O driver strength depend on the system implementation, it is difficult to determine these parameters in the standard host driver. When PVALEN is set to 1, automatic SDCLK frequency generation and driver strength selection are performed without considering system-specific conditions. This bit enables the functions defined in SDMMC_PVR.
If this bit is set to 0, SDMMC_HC2R.DRVSEL, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, SDMMC_HC2R.DRVSEL, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the SDMMC as specified in SDMMC_PVR.

Value	Description
0	SDCLK and Driver strength are controlled by the user.
1	Automatic selection by Preset Value is enabled.

Bit 14 – ASINTEN Asynchronous Interrupt Enable
This bit can be set to 1 if a card supports asynchronous interrupts and Asynchronous Interrupt Support (ASINTSUP) is set to 1 in SDMMC_CA0R. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode. If this bit is set to 1, the user can stop the SDCLK during the asynchronous interrupt period to save power. During this period, the SDMMC continues to deliver the Card Interrupt to the host when it is asserted by the card.

Value	Description
0	Disabled
1	Enabled

Bit 7 – SCLKSEL Sampling Clock Select
The SDMMC uses this bit to select the sampling clock to receive CMD and DAT.
This bit is set by the tuning procedure and is valid after completion of tuning (when EXTUN is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning has failed.
Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared by setting EXTUN to 1. Once the tuning circuit is reset, it takes time to complete the tuning sequence. Therefore, the user should keep this bit to 1 to perform a retuning sequence to complete a retuning sequence in a short time.
Changing this bit is not allowed while the SDMMC is receiving a response or a read data block. See Figure 2.29 in the “SD Host Controller Simplified Specification V3.00”.

Value	Description
0	The fixed clock is used to sample data.
1	The tuned clock is used to sample data.

Bit 6 – EXTUN Execute Tuning
This bit is set to 1 to start the tuning procedure and is automatically cleared when the tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select (SCLKSEL). The tuning procedure is aborted by writing 0. See Figure 2.29 in the “SD Host Controller Simplified Specification V3.00”.

Value	Description
0	Not tuned or tuning completed.
1	Execute tuning.

Bits 5:4 – DRVSEL[1:0] Driver Strength Select
The SDMMC output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set according to the Driver Type A, C and D support bits in SDMMC_CA1R.
This field depends on the setting of Preset Value Enable (PVALEN):
– PVALEN = 0: This field is set by the user.
– PVALEN = 1: This field is automatically set by a value specified in one of the SDMMC_PVRx.

Value	Name	Description
0	TYPEB	Driver Type B is selected (Default)
1	TYPEA	Driver Type A is selected
2	TYPEC	Driver Type C is selected
3	TYPED	Driver Type D is selected

Bit 3 – VS18EN 1.8V Signaling Enable
This bit controls the SDMMC_1V8SEL output, which in turn may control an external voltage regulator for the I/O cell and card I/Os. 3.3V or some other fixed voltage is supplied to the card/device regardless of the signaling voltage.
Setting this bit from 0 to 1 starts changing the signal voltage from 3.3V to 1.8V. The 1.8V regulator output must be stable within 5 ms.
Clearing this bit from 1 to 0 starts changing the signal voltage from 1.8V to 3.3V. The 3.3V regulator output must be stable within 5 ms.
The user can set this bit to 1 when the SDMMC supports 1.8V signaling (one of the support bits is set to 1: SDR50SUP, SDR104SUP or DDR50SUP in SDMMC_CA1R) and the card or device supports UHS-I (S18A = 1. See “Bus Switch Voltage Switch Sequence in the “Physical Layer Simplified Specification V3.01”).

Value	Description
0	3.3V signaling.
1	1.8V signaling.

Bits 2:0 – UHSMS[2:0] UHS Mode Select
This field is used to select one of the UHS-I modes and is effective when 1.8V Signal Enable (VS18EN) is set to 1. This field is effective only if SDMMC_MC1R.DDR is set to 0.
If Preset Value Enable is set to 1, the SDMMC sets SDCLK Frequency Select (SDCLKFSEL), Clock Generator Select (CLKGSEL) in SDMMC_CCR and Driver Strength Select (DRVSEL) according to SDMMC_PVR. In this case, one of the preset value registers is selected by this field. The user needs to reset SD Clock Enable (SDCLKEN) before changing this field to avoid generating a clock glitch. After setting this field, the user sets SDCLKEN to 1 again.

Value	Name	Description
0	SDR12	UHS SDR12 Mode
1	SDR25	UHS SDR25 Mode
2	SDR50	UHS SDR50 Mode
3	SDR104	UHS SDR104 Mode
4	DDR50	UHS DDR50 Mode

65.14.33 SDMMC Host Control 2 Register (e.MMC)

Name: SDMMC_HC2R (e.MMC)
Offset: 0x3E
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
	PVALEN							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	SCLKSEL	EXTUN	DRVSEL[1:0]		HS200EN[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – PVALEN Preset Value Enable
As the operating SDCLK frequency and I/O driver strength depend on the system implementation, it is difficult to determine these parameters in the standard host driver. When Preset Value Enable (PVALEN) is set to 1, automatic SDCLK frequency generation and driver strength selection are performed without considering system-specific conditions. This bit enables the functions defined in SDMMC_PVR.
If this bit is set to 0, SDMMC_HC2R.DRVSEL, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, SDMMC_HC2R.DRVSEL, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the SDMMC as specified in SDMMC_PVR.

Value	Description
0	SDCLK and Driver strength are controlled by the user.
1	Automatic selection by Preset Value is enabled.

Bit 7 – SCLKSEL Sampling Clock Select
The SDMMC uses this bit to select the sampling clock to receive CMD and DAT.
This bit is set by the tuning procedure and is valid after completion of tuning (when EXTUN is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning has failed.
Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared by setting EXTUN to 1. Once the tuning circuit is reset, it takes time to complete a tuning sequence. Therefore, the user should keep this bit to 1 to perform a retuning sequence to complete a retuning sequence in a short time. Changing this bit is not allowed while the SDMMC is receiving a response or a read data block. See Figure 2.29 in the “SD Host Controller Simplified Specification V3.00”.

Value	Description
0	The fixed clock is used to sample data.
1	The tuned clock is used to sample data.

Bit 6 – EXTUN Execute Tuning
This bit is set to 1 to start the tuning procedure and is automatically cleared when the tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select (SCLKSEL). The tuning procedure is aborted by writing 0. See Figure 2.29 in the “SD Host Controller Simplified Specification V3.00”.

Value	Description
0	Not tuned or tuning completed
1	Execute tuning

Bits 5:4 – DRVSEL[1:0] Driver Strength Select
The SDMMC output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set according to the Driver Type A, C and D support bits in SDMMC_CA1R.
This field depends on setting of Preset Value Enable (PVALEN):

– PVALEN = 0: This field is set by the user.
– PVALEN = 1: This field is automatically set by a value specified in one of the SDMMC_PVRx.

Value	Name	Description
0	TYPEB	Driver Type B is selected (Default)
1	TYPEA	Driver Type A is selected
2	TYPEC	Driver Type C is selected
3	TYPED	Driver Type D is selected

Bits 3:0 – HS200EN[3:0] HS200 Mode Enable
This field is used to select the e.MMC HS200 mode. When HS200EN is set to $B_{(hexa)}$, the HS200 mode is enabled. Any other value except 0 is forbidden when interfacing an e.MMC device.
If Preset Value Enable is set to 1, SDMMC sets SDCLK Frequency Select (SDCLKFSEL), Clock Generator Select (CLKGSEL) in SDMMC_CCR and Driver Strength Select (DRVSEL) according to SDMMC_PVR. In this case, one of the preset value registers is selected by this field. The user needs to reset SD Clock Enable (SDCLKEN) before changing this field to avoid generating a clock glitch. After setting this field, the user sets SDCLKEN to 1 again.
Note: This field is effective only if DDR in SDMMC_MC1R is set to 0.

65.14.34 SDMMC Capabilities 0 Register

Name: SDMMC_CA0R
Offset: 0x40
Property: Read/Write

Note: The reset values match the capabilities of the MPU alone. The user should adjust the capability registers so that they also match board design. Modify preset values only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Reset: The register reset values depend on the instance of the SDMMC:

Instance	Reset Value
SDMMC0	0x27EC6499
SDMMC1, SDMMC2	0x27E86499

Bit	31	30	29	28	27	26	25	24
	SLTYPE[1:0]		ASINTSUP	SB64SUP		V18VSUP	V30VSUP	V33VSUP
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP	MAXBLKL[1:0]	
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	BASECLKF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	TEOCLKU		TEOCLKF[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 31:30 – SLTYPE[1:0] Slot Type
This field indicates usage of a slot by a specific system. An SDMMC control register set is defined per slot. Embedded Slot for One Device means that only one nonremovable device is connected to a bus slot.
The Standard Host Driver controls a removable card (SLTYPE = 0) or one embedded device (SLTYPE = 1) connected to an SD bus slot.

Value	Name	Description
0	REMOVABLECARD	Removable Card Slot
1	EMBEDDED	Embedded Slot for One Device
2	–	Reserved
3	–	Reserved

Bit 29 – ASINTSUP Asynchronous Interrupt Support
See section “Asynchronous Interrupt” in the “SDIO Simplified Specification V3.00” .

Value	Description
0	Asynchronous interrupt not supported.
1	Asynchronous interrupt supported.

Bit 28 – SB64SUP 64-Bit System Bus Support
Reading this bit to 1 means that the SDMMC supports the 64-bit Address Descriptor mode and is connected to the 64-bit address system bus.

Value	Description
0	64-bit address bus not supported.
1	64-bit address bus supported.

Bit 26 – V18VSUP Voltage Support 1.8V

Value	Description
0	1.8V voltage supply not supported.
1	1.8V voltage supply supported.

Bit 25 – V30VSUP Voltage Support 3.0V

Value	Description
0	3.0V voltage supply not supported.
1	3.0V voltage supply supported.

Bit 24 – V33VSUP Voltage Support 3.3V

Value	Description
0	3.3V voltage supply not supported.
1	3.3V voltage supply supported.

Bit 23 – SRSUP Suspend/Resume Support
This bit indicates whether the SDMMC supports the Suspend/Resume functionality. If this bit is set to 0, the user does not issue either Suspend or Resume commands because the Suspend and Resume mechanism (see “Suspend and Resume Mechanism” in the “SD Host Controller Simplified Specification V3.00”) is not supported.

Value	Description
0	Suspend/Resume not supported.
1	Suspend/Resume supported.

Bit 22 – SDMASUP SDMA Support
This bit indicates whether the SDMMC is capable of using SDMA to transfer data between system memory and the SDMMC directly.

Value	Description
0	SDMA not supported.
1	SDMA supported.

Bit 21 – HSSUP High Speed Support
This bit indicates whether the SDMMC and the system support High Speed mode and they can supply SDCLK frequency from 25 MHz to 50 MHz.

Value	Description
0	High Speed not supported.
1	High Speed supported.

Bit 19 – ADMA2SUP ADMA2 Support
This bit indicates whether the SDMMC is capable of using ADMA2.

Value	Description
0	ADMA2 not supported.
1	ADMA2 supported.

Bit 18 – ED8SUP 8-Bit Support for Embedded Device
This bit indicates whether the SDMMC is capable of using the 8-bit Bus Width mode.

Value	Description
0	8-bit bus width not supported.
1	8-bit bus width supported.

Bits 17:16 – MAXBLKL[1:0] Max Block Length
This field indicates the maximum block size that the user can read and write to the buffer in the SDMMC. Three sizes can be defined, as shown below. It is noted that the transfer block length is always 512 bytes for SD Memory Cards regardless of this field.

Value	Name	Description
0	512	512 bytes
1	1024	1024 bytes
2	2048	2048 bytes
3	NONE	Reserved

Bits 15:8 – BASECLKF[7:0] Base Clock Frequency
This value indicates the frequency of the base clock (BASECLK). The user uses this value to calculate the clock divider value (see SDCLK Frequency Select (SDCLKFSEL) in SDMMC_CCR).
If this field is set to 0, the user must get the information via another method.
$f_{\mathrm{BASECLK}} = \mathrm{BASECLKF}_{\mathrm{MHz}}$

Bit 7 – TEOCLKU Timeout Clock Unit
This bit shows the unit of the base clock frequency used to detect Data Timeout Error.

Value	Description
0	KHz
1	MHz

Bits 5:0 – TEOCLKF[5:0] Timeout Clock Frequency
This bit shows the timeout clock frequency (TEOCLK) used to detect Data Timeout Error.
If this field is set to 0, the user must get the information via another method.
The Timeout Clock Unit (TEOCLKU) defines the unit of this field’s value.
– TEOCLKU = 0: $f_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{KHz}}$
– TEOCLKU = 1: $f_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{MHz}}$

65.14.35 SDMMC Capabilities 1 Register

Name: SDMMC_CA1R
Offset: 0x44
Reset: 0x00010F77
Property: Read/Write

Note: The reset values match the capabilities of the MPU alone. The user should adjust the capability registers so that they also match board design. Modify preset values only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CLKMULT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	RTMOD[1:0]		TSDR50		TCNTRT[3:0]			
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		1	1	1	1

Bit	7	6	5	4	3	2	1	0
		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	1	1		1	1	1

Bits 23:16 – CLKMULT[7:0] Clock Multiplier
This field indicates the multiplier factor between the Base Clock (BASECLK) used for the Divided Clock Mode and the Multiplied Clock (MULTCLK) used for the Programmable Clock mode (see SDMMC_CCR).
Reading this field to 0 means that the Programmable Clock mode is not supported.
$f_{\mathrm{MULTCLK}} = f_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT} + 1)$

Bits 15:14 – RTMOD[1:0] Retuning Modes
This field selects the retuning method and limits the maximum data length.
There are two retuning timings: Retuning Request (RTREQ) controlled by the SDMMC, and expiration of a Retuning timer controlled by the user. By receiving either timing, the user executes the retuning procedure just before a next command issue.
The maximum data length per read/write command is restricted so that retuning procedures can be inserted during data transfers.
Retuning Mode 1:
The SDMMC does not have any internal logic to detect when retuning needs to be performed. In this case, the user should maintain all retuning timings by using the Retuning Timer. To enable inserting the retuning procedure during data transfers, the data length per Read/Write command must be limited to 4 Mbytes.
Retuning Mode 2:
The SDMMC has the capability to indicate the retuning timing by Retuning Request (RTREQ) during data transfers. Then the data length per Read/Write command must be limited to 4 Mbytes.
During nondata transfer, retuning timing is determined by either Retuning Request or Retuning Timer. If Retuning Request is used, Retuning Timer should be disabled.
Retuning Mode 3:
The SDMMC has the capability to take care of the retuning during data transfer (Auto Retuning). Retuning Request is not generated during data transfers and there is no limitation to data length per Read/Write command.

During nondata transfer, retuning timing is determined either by Retuning Request or Retuning Timer. If Retuning Request is used, Retuning Timer should be disabled.

Value	Name	Description	Data Length
0	MODE1	Timer	4 Mbytes (Max)
1	MODE2	Timer and Retuning Request	4 Mbytes (Max)
2	MODE3	Auto Retuning (for transfer) Timer and Retuning Request	Any
3	–	Reserved	–

Bit 13 – TSDR50 Use Tuning for SDR50

If this bit is set to 1, the SDMMC requires tuning to operate SDR50 (tuning is always required to operate SDR104).

Value	Description
0	SDR50 does not require tuning.
1	SDR50 requires tuning.

Bits 11:8 – TCNTRT[3:0] Timer Count For Retuning

This field indicates an initial value of the Retuning Timer for Retuning Mode (RTMOD) 1 to 3. Reading this field at 0 means that the Retuning Timer is disabled. The Retuning Timer initial value ranges from 0 to 1024 seconds.

$t_{TIMER} = 2^{(TCNTRT - 1)}$Seconds

Bit 6 – DRVDSUP Driver Type D Support

Value	Description
0	Driver type D is not supported.
1	Driver type D is supported.

Bit 5 – DRVCSUP Driver Type C Support

Value	Description
0	Driver type C is not supported.
1	Driver type C is supported.

Bit 4 – DRVASUP Driver Type A Support

Value	Description
0	Driver type A is not supported.
1	Driver type A is supported.

Bit 2 – DDR50SUP DDR50 Support

Value	Description
0	DDR50 mode is not supported.
1	DDR50 mode is supported.

Bit 1 – SDR104SUP SDR104 Support

Value	Description
0	SDR104 mode is not supported.
1	SDR104 mode is supported.

Bit 0 – SDR50SUP SDR50 Support

Value	Description
0	SDR50 mode is not supported.
1	SDR50 mode is supported.

65.14.36 SDMMC Maximum Current Capabilities Register

Name: SDMMC_MCCAR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Note: The user should adjust the Maximum Current Capabilities register so that it matches board design. The preset values can be modified only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	MAXCUR18V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MAXCUR30V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MAXCUR33V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – MAXCUR18V[7:0] Maximum Current for 1.8V
This field indicates the maximum current capability for 1.8V voltage. This value is meaningful only if V18VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR18V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR18}V$

Bits 15:8 – MAXCUR30V[7:0] Maximum Current for 3.0V
This field indicates the maximum current capability for 3.0V voltage. This value is meaningful only if V30VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR30V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR30}V$

Bits 7:0 – MAXCUR33V[7:0] Maximum Current for 3.3V
This field indicates the maximum current capability for 3.3V voltage. This value is meaningful only if V33VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR33V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR33}V$

65.14.37 SDMMC Force Event Register for Auto CMD Error Status

Name: SDMMC_FERACES
Offset: 0x50
Reset: –
Property: Write-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	W			W	W	W	W	W
Reset	–			–	–	–	–	–

Bit 7 – CMDNI Force Event for Command Not Issued by Auto CMD12 Error
For test purposes, the user can write this bit to 1 to raise the CMDNI status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 4 – ACMDIDX Force Event for Auto CMD Index Error
For test purposes, the user can write this bit to 1 to raise the ACMDIDX status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 3 – ACMDEND Force Event for Auto CMD End Bit Error
For test purposes, the user can write this bit to 1 to raise the ACMDEND status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 2 – ACMDCRC Force Event for Auto CMD CRC Error
For test purposes, the user can write this bit to 1 to raise the ACMDCRC status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 1 – ACMDTEO Force Event for Auto CMD Timeout Error
For test purposes, the user can write this bit to 1 to raise the ACMDTEO status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 0 – ACMD12NE Force Event for Auto CMD12 Not Executed
For test purposes, the user can write this bit to 1 to raise the ACMD12NE status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

65.14.38 SDMMC Force Event Register for Error Interrupt Status

Name: SDMMC_FEREIS
Offset: 0x52
Reset: –
Property: Write-only

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access				W			W	W
Reset				–			–	–

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 12 – BOOTAE Force Event for Boot Acknowledge Error
For test purposes, the user can write this bit to 1 to raise the BOOTAE status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 9 – ADMA Force Event for ADMA Error
For test purposes, the user can write this bit to 1 to raise the ADMA status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 8 – ACMD Force Event for Auto CMD Error
For test purposes, the user can write this bit to 1 to raise the ACMD status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 7 – CURLIM Force Event for Current Limit Error
For test purposes, the user can write this bit to 1 to raise the CURLIM status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 6 – DATEND Force Event for Data End Bit Error
For test purposes, the user can write this bit to 1 to raise the DATEND status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 5 – DATCRC Force Event for Data CRC error
For test purposes, the user can write this bit to 1 to raise the DATCRC status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 4 – DATTEO Force Event for Data Timeout error
For test purposes, the user can write this bit to 1 to raise the DATTEO status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 3 – CMDIDX Force Event for Command Index Error
For test purposes, the user can write this bit to 1 to raise the CMDIDX status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 2 – CMDEND Force Event for Command End Bit Error
For test purposes, the user can write this bit to 1 to raise the CDMEND status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 1 – CMDCRC Force Event for Command CRC Error
For test purposes, the user can write this bit to 1 to raise the CMDCRC status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 0 – CMDTEO Force Event for Command Timeout Error
For test purposes, the user can write this bit to 1 to raise the CMDTEO status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

65.14.39 SDMMC ADMA Error Status Register

Name: SDMMC_AESR
Offset: 0x54
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
						LMIS	ERRST[1:0]	
Access						R	R	R
Reset						0	0	0

Bit 2 – LMIS ADMA Length Mismatch Error
This error occurs in the following two cases:
– While Block Count Enable (BCEN) is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count (BLKCNT) and Transfer Block Size (BLKSIZE).
– The total data length cannot be divided by the Transfer Block Size (BLKSIZE).

Value	Description
0	No error
1	Error

Bits 1:0 – ERRST[1:0] ADMA Error State
This field indicates the state of ADMA when an error has occurred during an ADMA data transfer. This field never reads 2 because ADMA never stops in this state.

Value	Name	Description
0	STOP	(Stop DMA) SDMMC_ASAR points to the descriptor following the error descriptor
1	FDS	(Fetch Descriptor) SDMMC_ASAR points to the error descriptor
2	–	(Not used)
3	TFR	(Transfer Data) SDMMC_ASAR points to the descriptor following the error descriptor

65.14.40 SDMMC ADMA System Address Register 0

Name: SDMMC_ASAR0
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADMASA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADMASA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADMASA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADMASA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADMASA[31:0] ADMA System Address
This field holds the byte address of the executing command of the descriptor table. The 32-bit address descriptor uses SDMMC_ASAR. At the start of ADMA, the user must set the start address of the descriptor table. The ADMA increments this register address, which points to the next Descriptor line to be fetched.
When the ADMA Error (ADMA) status flag rises, this field holds a valid descriptor address depending on the ADMA Error State (ERRST). The user must program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 0.

65.14.41 SDMMC Preset Value Register

Name: SDMMC_PVRx
Offset: 0x60 + x*0x02 [x=0..7]
Reset: 0x00000000
Property: Read/Write

One of the Preset Value registers is effective based on the selected bus speed mode. The table below defines the conditions to select one of the SDMMC_PVRs.

Table 65-5. Preset Value Register Select Condition

Selected Bus Speed Mode	VS18EN (SDMMC_HC2R)	HSEN (SDMMC_HC1R)	UHSMS (SDMMC_HC2R)
Default Speed	0	0	don’t care
High Speed	0	1	don’t care
SDR12	1	don’t care	0
SDR25	1	don’t care	1
SDR50	1	don’t care	2
SDR104/HS200	1	don’t care	3
DDR50	1	don’t care	4
Reserved	1	don’t care	Other values

The table below shows the effective Preset Value Register according to the Selected Bus Speed mode.

Table 65-6. Preset Value Registers

SDMMC_PVRx	Selected Bus Speed Mode	Signal Voltage
SDMMC_PVR0	Initialization	3.3V
SDMMC_PVR1	Default Speed	3.3V
SDMMC_PVR2	High Speed	3.3V
SDMMC_PVR3	SDR12	1.8V
SDMMC_PVR4	SDR25	1.8V
SDMMC_PVR5	SDR50	1.8V
SDMMC_PVR6	SDR104/HS200	1.8V
SDMMC_PVR7	DDR50	1.8V

When Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, Driver Strength Select (DRVSEL) in SDMMC_HC2R and SDCLK Frequency Select (SDLCKFSEL) and Clock Generator Select (CLKGSEL) in SDMMC_CCR are automatically set based on the Selected Bus Speed mode. This means that the user does not need to set these fields when preset is enabled. A Preset Value Register for Initialization (SDMMC_PVR0) is not selected by Bus Speed mode. Before starting the initialization sequence, the user needs to set a clock preset value to SDCLKFSEL in SDMMC_CCR.PVALEN can be set to 1 after the initialization is completed.

Bit	15	14	13	12	11	10	9	8
	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	SDCLKFSEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:14 – DRVSEL[1:0] Driver Strength Select
See DRVSEL in SDMMC_HC2R (SD_SDIO) / SDMMC_HC2R (e.MMC).

Bit 10 – CLKGSEL Clock Generator Select
See CLKGSEL in SDMMC_CCR.

Bits 9:0 – SDCLKFSEL[9:0] SDCLK Frequency Select
See SDCLKFSEL in SDMMC_CCR.

65.14.42 SDMMC Slot Interrupt Status Register

Name: SDMMC_SISR
Offset: 0xFC
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						INTSSL[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – INTSSL[2:0] Interrupt Signal for Each Slot
These status bits indicate the logical OR of Interrupt Signals and Wakeup Signal for each SDMMC instance in the product (INTSSL[x] corresponds to SDMMCx instance in the product).

65.14.43 SDMMC Host Controller Version Register

Name: SDMMC_HCVR
Offset: 0xFE
Reset: 0x3002
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	VVER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SVER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	0

Bits 15:8 – VVER[7:0] Vendor Version Number
Reserved. Value subject to change. No functionality associated. This is the internal version of the module.

Bits 7:0 – SVER[7:0] Specification Version Number
This status indicates the SD Host Controller Specification Version.

Value	Name
0	SD Host Specification Version 1.00
1	SD Host Specification Version 2.00, including the feature of the ADMA and Test Register
2	SD Host Specification Version 3.00

65.14.44 SDMMC Additional Present State Register

Name: SDMMC_APSR
Offset: 0x200
Reset: –
Property: Read-only

Register reset value:

Instance	Reset Value
SDMMC0, SDMMC1, SDMMC2	0x0000000F

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					HDATLL[3:0]			
Access					R	R	R	R
Reset					–	–	–	–

Bits 3:0 – HDATLL[3:0] DAT[7:4] High Line Level
This status is used to check the DAT[7:4] line level to recover from errors, and for debugging.

65.14.45 SDMMC e.MMC Control 1 Register

Name: SDMMC_MC1R
Offset: 0x204
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
	FCD	RSTN	BOOTA	OPD	DDR		CMDTYP[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 7 – FCD e.MMC Force Card Detect
When using e.MMC, the user can set this bit to 1 to bypass the card detection procedure using the SDMMC_CD signal.
0 (DISABLED): e.MMC Forced Card Detect is disabled. The SDMMC_CD signal is used and debounce timing is applied.
1 (ENABLED): e.MMC Forced Card Detect is enabled.

Bit 6 – RSTN e.MMC Reset Signal
This bit controls the e.MMC reset signal.

Value	Description
0	Reset signal is inactive.
1	Reset signal is active.

Bit 5 – BOOTA e.MMC Boot Acknowledge Enable
This bit must be set according to the value of BOOT_ACK in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).
When this bit is set to 1, the SDMMC waits for boot acknowledge pattern from the e.MMC before receiving boot data.
If the boot acknowledge pattern is wrong, the BOOTAE status flag rises in SDMMC_EISTR if BOOTAE is set in SDMMC_EISTER. An interrupt is generated if BOOTAE is set in SDMMC_EISIER.
If the no boot acknowledge pattern is received, the DATTEO status flag rises in SDMMC_EISTR if DATTEO is set in SDMMC_EISTER. An interrupt is generated if DATTEO is set in SDMMC_EISIER.

Bit 4 – OPD e.MMC Open Drain Mode
This bit sets the command line in open drain.

Value	Description
0	The command line is in push-pull.
1	The command line is in open drain.

Bit 3 – DDR e.MMC HSDDR Mode
This bit selects the High Speed DDR mode.
The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when DDR is 1.

Value	Description
0	High Speed DDR is not selected.
1	High Speed DDR is selected.

Bits 1:0 – CMDTYP[1:0] e.MMC Command Type

Value	Name	Description
0	NORMAL	The command is not an e.MMC specific command.
1	WAITIRQ	This bit must be set to 1 when the e.MMC is in Interrupt mode (CMD40). See “Interrupt Mode” in the “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .
2	STREAM	This bit must be set to 1 in the case of Stream Read(CMD11) or Stream Write (CMD20). Only effective for e.MMC up to revision 4.41.
3	BOOT	Starts a Boot Operation mode at the next write to SDMMC_CR. Boot data are read directly from e.MMC device.

65.14.46 SDMMC e.MMC Control 2 Register

Name: SDMMC_MC2R
Offset: 0x205
Reset: –
Property: Write-only

Bit	7	6	5	4	3	2	1	0
							ABOOT	SRESP
Access							W	W
Reset							–	–

Bit 1 – ABOOT e.MMC Abort Boot
This bit is used to exit from Boot mode. Writing this bit to 1 exits the Boot Operation mode. Writing 0 is ignored.

Bit 0 – SRESP e.MMC Abort Wait IRQ
This bit is used to exit from the Interrupt mode. When this bit is written to 1, the SDMMC sends the CMD40 response automatically. This brings the e.MMC from Interrupt mode to the standard Data Transfer mode. Writing this bit to 0 is ignored.
This bit is only effective when CMD_TYP in SDMMC_MC1R is set to WAITIRQ.

65.14.47 SDMMC e.MMC Control 3 Register

Name: SDMMC_MC3R
Offset: 0x206
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
			DQSUPVAL[2:0]				ESMEN	HS400EN
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bits 5:3 – DQSUPVAL[2:0] DQS Delay Update Timer Value
0: DQS delay update is performed each time a command is sent to the device.
n: DQS delay update is performed before sending the command only if a timer delay is elapsed. The timer delay t is defined as follows:
$t = \left(2^{n+1} - 1\right) \times t_{\mathrm{SLCK}}$

Bit 1 – ESMEN Enhanced Strobe Mode Enable
This field is used to enable the Enhanced Strobe Mode. User must ensure that this mode is supported by the e.MMC device (Only devices compliant with the Embedded Multimedia Card (e.MMC) Electrical Standard 5.1 and further support this mode).
0 (DISABLED): Enhanced Strobe mode is disabled.
1 (ENABLED): Enhanced Strobe mode is enabled.

Bit 0 – HS400EN HS400 Mode Enable
This field is used to select the e.MMC HS400 mode.
0 (DISABLED): HS400 mode is disabled.
1 (ENABLED): HS400 mode is enabled.
If Preset Value Enable is set to 1, SDMMC sets SDCLK Frequency Select (SDCLKFSEL), Clock Generator Select (CLKGSEL) in SDMMC_CCR and Driver Strength Select (DRVSEL) according to SDMMC_PVR. In this case, SDMMC_EPVR8 register is selected by this bit. The user needs to reset SD Clock Enable (SDCLKEN) before changing this field to avoid generating a clock glitch. After setting this field, the user sets SDCLKEN to 1 again.

65.14.48 SDMMC Debounce Register

Name: SDMMC_DEBR
Offset: 0x207
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
							CDDVAL[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – CDDVAL[1:0] Card Detect Debounce Value
Defines the debounce delay for card insertion/removal.

CDDVAL	Number of Slow Clock Cycles N
0	1
1	8
2	33
3	328

$$\text{Debounce} = N \times t_{\text{SLCK}}$$

65.14.49 SDMMC AHB Control Register

Name: SDMMC_ACR
Offset: 0x208
Reset: 0x00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DFQOS[3:0]						BUFM[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
							BMAX[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 15:12 – DFQOS[3:0] Descriptor Fetch QOS
This field defines the QOS value of ADMA AHB access when fetching descriptor. For all other accesses, the QOS is set to 0.

Bits 9:8 – BUFM[1:0] AHB Bufferable Mode
This field defines if last access of data transfer is bufferable or not.

Value	Name	Description
0	NEVER	All SDMA/ADMA AHB accesses are not bufferable.
1	ALWAYS	All SDMA/ADMA AHB accesses are bufferable.
2	BLOCK	All SDMA/ADMA AHB accesses are bufferable except the last access of a data block.
3	TRANSFER	All SDMA/ADMA AHB accesses are bufferable except the last access of a data transfer.

Bits 1:0 – BMAX[1:0] AHB Maximum Burst
This field selects the maximum burst size in case of DMA transfer.

Value	Name	Description
0	INCR16	The maximum burst size is INCR16.
1	INCR8	The maximum burst size is INCR8.
2	INCR4	The maximum burst size is INCR4.
3	SINGLE	Only SINGLE transfers are performed.

65.14.50 SDMMC Clock Control 2 Register

Name: SDMMC_CC2R
Offset: 0x20C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FSDCLKD
Access								R/W
Reset								0

Bit 0 – FSDCLKD Force SDCLK Disabled
The user can choose to maintain the SDCLK during 8 SDCLK cycles after the end bit of the last data block in case of a read transaction, or after the end bit of the CRC status in case of a write transaction.

Value	Description
0	The SDCLK is forced and it cannot be stopped immediately after the transaction.
1	The SDCLK is not forced and it can be stopped immediately after the transaction.

65.14.51 SDMMC Retuning Control 1 Register

Name: SDMMC_RTC1R
Offset: 0x210
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								TMREN
Access								R/W
Reset								0

Bit 0 – TMREN Retuning Timer Enable
Enable the retuning timer.
0 (DISABLED): The retuning timer is disabled.
1 (ENABLED): The retuning timer is enabled.

65.14.52 SDMMC Retuning Control 2 Register

Name: SDMMC_RTC2R
Offset: 0x211
Reset: –
Property: Write-only

Bit	7	6	5	4	3	2	1	0
								RLD
Access								W
Reset								–

Bit 0 – RLD Retuning Timer Reload
This bit is only efficient if the Retuning timer is enabled (SDMMC_RTC1R.TMREN set to 1). Once the Timer Counter Value (TCVAL) is set to a nonzero value in SDMMC_RTCVR, setting this bit to 1 starts the timer count. The retuning timer count restarts each time this bit is written to 1.
Writing this bit to 0 has no effect.

65.14.53 SDMMC Retuning Counter Value Register

Name: SDMMC_RTCVR
Offset: 0x214
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TCVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – TCVAL[3:0] Retuning Timer Counter Value
The TCVAL value is used to define the time before expiration of the retuning timer where:

$$\mathrm{Time} = 2^{\mathrm{TCVAL} - 1}\mathrm{seconds}$$

This value must range between 1 and 11. Any other value results in the retuning timer being disabled.

65.14.54 SDMMC Retuning Interrupt Status Enable Register

Name: SDMMC_RTISTER
Offset: 0x218
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								TEVT
Access								R/W
Reset								0

Bit 0 – TEVT Retuning Timer Event
0 (MASKED): The TEVT status flag in SDMMC_RTISTR is masked.
1 (ENABLED): The TEVT status flag in SDMMC_RTISTR is enabled.

65.14.55 SDMMC Retuning Interrupt Signal Enable Register

Name: SDMMC_RTISIER
Offset: 0x219
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								TEVT
Access								R/W
Reset								0

Bit 0 – TEVT Retuning Timer Event
0 (MASKED): No interrupt is generated when the TEVT status rises in SDMMC_RTISTR.
1 (ENABLED): An interrupt is generated when the TEVT status rises in SDMMC_RTISTR.

65.14.56 SDMMC Retuning Interrupt Status Register

Name: SDMMC_RTISTR
Offset: 0x21C
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								TEVT
Access								R/W
Reset								0

Bit 0 – TEVT Retuning Timer Event
This bit is set to 1 when the retuning timer count is elapsed if SDMMC_RTISTER.TEVT is set to 1. An interrupt is generated if SDMMC_RTISIER.TEVT is set to 1.
Writing this bit to 1 clears the bit.

Value	Description
0	No retuning timer event.
1	Retuning timer event.

65.14.57 SDMMC Retuning Status Slots Register

Name: SDMMC_RTSSR
Offset: 0x21D
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
						TEVTSLOT[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – TEVTSLOT[2:0] Retuning Timer Event Slots
Indicates the TEVT status for each SDMMC instance in the product (TEVTSLOT[x] corresponds to SDMMCx instance in the product).

65.14.58 SDMMC Tuning Control Register

Name: SDMMC_TUNCR
Offset: 0x220
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SMPLPT
Access								R/W
Reset								0

Bit 0 – SMPLPT Sampling Point
This bit selects the position of the sampling point into the data window for SDR104 and HS200 modes.

Value	Description
0	Sampling point is set at 50% of the data window.
1	Sampling point is set at 75% of the data window.

65.14.59 SDMMC Capabilities Control Register

Name: SDMMC_CACR
Offset: 0x230
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								CAPWREN
Access								R/W
Reset								0

Bits 15:8 – KEY[7:0] Key

Value	Name	Description
0x46	KEY	Writing any other value in this field aborts the write operation of the CAPWREN bit. Always reads as 0.

Bit 0 – CAPWREN Capabilities Write Enable
This bit can only be written if the value of KEY corresponds to 0x46.

Value	Description
0	Capabilities registers (SDMMC_CA0R, SDMMC_CA1R and SDMMC_CA1R) cannot be written.
1	Capabilities registers (SDMMC_CA0R, SDMMC_CA1R and SDMMC_CA1R) can be written.

65.14.60 SDMMC Debug Register

Name: SDMMC_DBGR
Offset: 0x234
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NIDBG
Access								R/W
Reset								0

Bit 0 – NIDBG Nonintrusive Debug
0 (DISABLED): Reading the SDMMC_BDPR via debugger increments the dual port RAM read pointer.
1 (ENABLED): Reading the SDMMC_BDPR via debugger does not increment the dual port RAM read pointer.

65.14.61 SDMMC Calibration Control Register

Name: SDMMC_CALCR
Offset: 0x240
Reset: 0x0000500E
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CALPBP[3:0]				CALP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CALNBP[3:0]				CALN[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CNTVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		BPEN	TUNDIS	ALWYSON	CLKDIV[2:0]			EN
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	1	1	1	0

Bits 31:28 – CALPBP[3:0] Calibration P Bypass Value
Calibration code applied for the p-channel transistors when BPEN is set to 1. This field is ignored if BPEN is 0.

Bits 27:24 – CALP[3:0] Calibration P Status
Calibration code for the p-channel transistors to match the required output impedance.

Bits 23:20 – CALNBP[3:0] Calibration N Bypass Value
Calibration code applied for the n-channel transistors when BPEN is set to 1. This field is ignored if BPEN is 0.

Bits 19:16 – CALN[3:0] Calibration N Status
Calibration code for the n-channel transistors to match the required output impedance.

Bits 15:8 – CNTVAL[7:0] Calibration Counter Value
Defines the number of XXXX cycles (divided by 4) required to cover the I/O calibration cell startup time.

$$\mathrm{CNTVAL}_{\mathrm{Minimum}} = \frac{t_{\mathrm{STARTUP}}}{4 \times t_{\mathrm{HCLOCK}}}$$

$t_{STARTUP}$ = 2 μs

Bit 6 – BPEN Calibration Bypass Enabled

Value	Description
0	Calibration bypass is not enabled.
1	Calibration bypass is enabled. CALPBP and CALNBP codes are applied to the calibration cell.

Bit 5 – TUNDIS Calibration During Tuning Disabled

Value	Description
0	Calibration is launched before each tuning.
1	Calibration is not launched at tuning.

Bit 4 – ALWYSON Calibration Analog Always ON

Value	Description
0	Calibration analog is shut down after each calibration.
1	Calibration analog remains powered after calibration.

Bits 3:1 – CLKDIV[2:0] Calibration Clock Division
The clock applied to the calibration cell is divided by CLKDIV + 1

Bit 0 – EN PADs Calibration Enable

Value	Description
0	SDMMC I/O calibration disabled.
1	SDMMC I/O calibration enabled.

65.14.62 SDMMC Extended Preset Value Register 8

Name: SDMMC_EPVR8
Offset: 0x244
Reset: 0x0000
Property: Read/Write

The Preset Value register 8 is effective based on the selected bus speed mode. The table below defines the conditions to select SDMMC_EPVR8.

Table 65-7. Preset Value Register Select Condition

Selected Bus Speed Mode	HS400EN (SDMMC_MC3R)
HS400	1

When Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, SDCLK Frequency Select (SDLCKFSEL) and Clock Generator Select (CLKGSEL) in SDMMC_CCR, and Driver Strength Select (DRVSEL) in SDMMC_HC2R are automatically set based on the selected bus speed mode. This means that the user does not need to set these fields when preset is enabled. Before starting the initialization sequence, the user needs to set a clock preset value to SDMMC_CCR.SDCLKFSEL. Bit PVALEN can be set to 1 after the initialization is completed.

Bit	15	14	13	12	11	10	9	8
	DRVSEL[1:0]					CLKGSEL	SDCLKFSEL[9:8]	
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	SDCLKFSEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:14 – DRVSEL[1:0] Driver Strength Select
See DRVSEL in SDMMC_HC2R.

Bit 10 – CLKGSEL Clock Generator Select
See CLKGSEL in SDMMC_CCR.

Bits 9:0 – SDCLKFSEL[9:0] SDCLK Frequency Select
See SDCLKFSEL in SDMMC_CCR.

66. Controller Area Network (MCAN)

66.1 Description

The Controller Area Network (MCAN) performs communication according to ISO 11898-1:2015 and to Bosch CAN FD specification. Additional transceiver hardware is required for connection to the physical layer.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN core to the Message RAM, as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN core, as well as providing transmit status information.

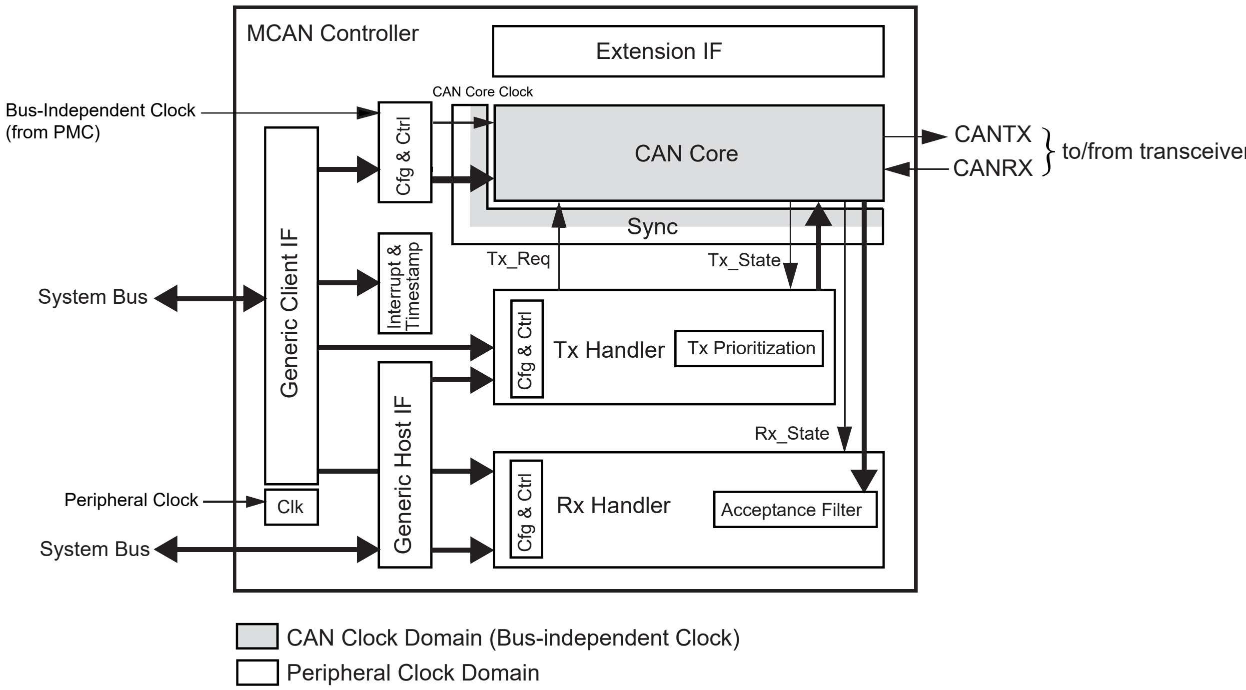
Acceptance filtering is implemented by a combination of up to 128 filter elements, where each element can be configured as a range, as a bit mask, or as a dedicated ID filter.

66.2 Embedded Characteristics

- Compliant with CAN Protocol Version 2.0 Part A, B and ISO 11898-1
- CAN FD with up to 64 Data Bytes Supported
- CAN Error Logging
- AUTOSAR Optimized
- SAE J1939 Optimized
- Improved Acceptance Filtering
- Two Configurable Receive FIFOs
- Separate Signalling on Reception of High Priority Messages
- Up to 64 Dedicated Receive Buffers
- Up to 32 Dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM Access for Processor
- Multiple MCANs May Share the Same Message RAM
- Programmable Loop-back Test Mode
- Maskable Module Interrupts
- Support for Asynchronous CAN and System Bus Clocks
- Power-down Support
- Debug on CAN Support

66.3 Block Diagram

Figure 66-1. MCAN Block Diagram



Note: Refer to section “Power Management Controller (PMC)” for details about the bus-independent clock (GCLK).

66.4 Product Dependencies

66.4.1 I/O Lines

The pins used to interface to the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the CAN pins to their peripheral functions.

66.4.2 Power Management

The MCAN can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the MCAN clock.

In order to achieve a stable function of the MCAN, the system bus clock must always be faster than or equal to the CAN clock.

It is recommended to use the CAN clock at frequencies of 20, 40 or 80 MHz. GCLK allows the system bus and processor clock to be modified without affecting the bit rate communication.

66.4.3 Interrupt Sources

The two MCAN interrupt lines (MCAN_INT0, MCAN_INT1) are connected on internal sources of the Interrupt Controller.

Using the MCAN interrupts requires the Interrupt Controller to be programmed first.

Interrupt sources can be routed either to MCAN_INT0 or to MCAN_INT1. By default, all interrupt sources are routed to interrupt line MCAN_INT0/1. By programming MCAN_ILE.EINT0 and MCAN_ILE.EINT1, the interrupt sources can be enabled or disabled separately.

66.4.4 Address Configuration

The LSBs [bits 15:2] for each section of the CAN Message RAM are configured in the respective buffer configuration registers as detailed in Message RAM.

The MSBs [bits 31:16] of the CAN Message RAM for CAN0 and CAN1 are configured at 0x00300000.

66.4.5 Timestamping

Timestamping uses the value of CV in the Timer Counter Channel 0 register (TC_CV0) of Timer Counter 1. Timer Counter 1 may use the generic clock (GCLK) from the Power Management Controller. Refer to the section "Timer Counter (TC)" for more details about clock source selection.

66.5 Functional Description

66.5.1 Operating Modes

66.5.1.1 Software Initialization

Software initialization is started by setting bit MCAN_CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While MCAN_CCCR.INIT is set, message transfer from and to the CAN bus is stopped and the status of the CAN bus output CANTX is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting MCAN_CCCR.INIT does not change any configuration register. Resetting MCAN_CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (≡ Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the MCAN configuration registers is only enabled when both bits MCAN_CCCR.INIT and MCAN_CCCR.CCE are set (protected write).

MCAN_CCCR.CCE can only be configured when MCAN_CCCR.INIT = ‘1’. MCAN_CCCR.CCE is automatically cleared when MCAN_CCCR.INIT = ‘0’.

The following registers are cleared when MCAN_CCCR.CCE = ‘1’:

- High Priority Message Status (MCAN_HPMS)
- Receive FIFO 0 Status (MCAN_RXF0S)
- Receive FIFO 1 Status (MCAN_RXF1S)
- Transmit FIFO/Queue Status (MCAN_TXFQS)
- Transmit Buffer Request Pending (MCAN_TXBRP)
- Transmit Buffer Transmission Occurred (MCAN_TXBTO)
- Transmit Buffer Cancellation Finished (MCAN_TXBCF)
- Transmit Event FIFO Status (MCAN_TXEFS)

The Timeout Counter value MCAN_TOCV.TOC is loaded with the value configured by MCAN_TOCC.TOP when MCAN_CCCR.CCE = ‘1’.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = ‘1’.

The following registers are only writeable while MCAN_CCCR.CCE = ‘0’

- Transmit Buffer Add Request (MCAN_TXBAR)
- Transmit Buffer Cancellation Request (MCAN_TXBCR)

MCAN_CCCR.TEST and MCAN_CCCR.MON can only be set when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’. Both bits may be cleared at any time. MCAN_CCCR.DAR can only be configured when MCAN_CCCR.INIT = ‘1’ and MCAN_CCCR.CCE = ‘1’.

66.5.1.2 Normal Operation

Once the MCAN is initialized and MCAN_CCCR.INIT is cleared, the MCAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted, dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

66.5.1.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit MCAN_PSR.PXE. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 2) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE. In case CCCR.FDOE = ‘1’, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = ‘0’, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only bit FDF of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting according to ISO11898-1 until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD-capable. Non-CAN FD nodes are held in Silent mode until programming has completed. Then all nodes revert to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to the table below.

Table 66-1. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing and Prescaler register (MCAN_NBTP). In the following CAN FD data phase, the data phase CAN bit timing is used as defined by the FastData Bit Timing and Prescaler register (MCAN_DBTP). The bit timing reverts back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN core clock frequency. Example: with a CAN clock frequency of 20 MHz and the shortest configurable bit time of 4 t_q, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

66.5.1.4 Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANTX the protocol controller receives the transmitted data from its local CAN transceiver via pin CANRX. The received data is delayed by the transmitter delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transmitter delay, the delay compensation is introduced. Without delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the delay.

66.5.1.4.1 Description

The MCAN protocol unit has implemented a delay compensation mechanism to compensate the delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase, the delayed transmit data is compared against the received data at the secondary sample point. If a bit error is detected, the transmitter will react to this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit MCAN_DBTP.TDC.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output CANTX through the transceiver to the receive input CANRX plus the transmitter delay compensation offset as configured by MCAN_TDCR.TDCO. The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of CAN core clock periods.

MCAN_PSR.TDCV shows the actual transmitter delay compensation value. MCAN_PSR.TDCV is cleared when MCAN_CCCR.INIT is set and is updated at each transmission of an FD frame while MCAN_DBTP.TDC is set.

The following boundary conditions have to be considered for the delay compensation implemented in the MCAN:

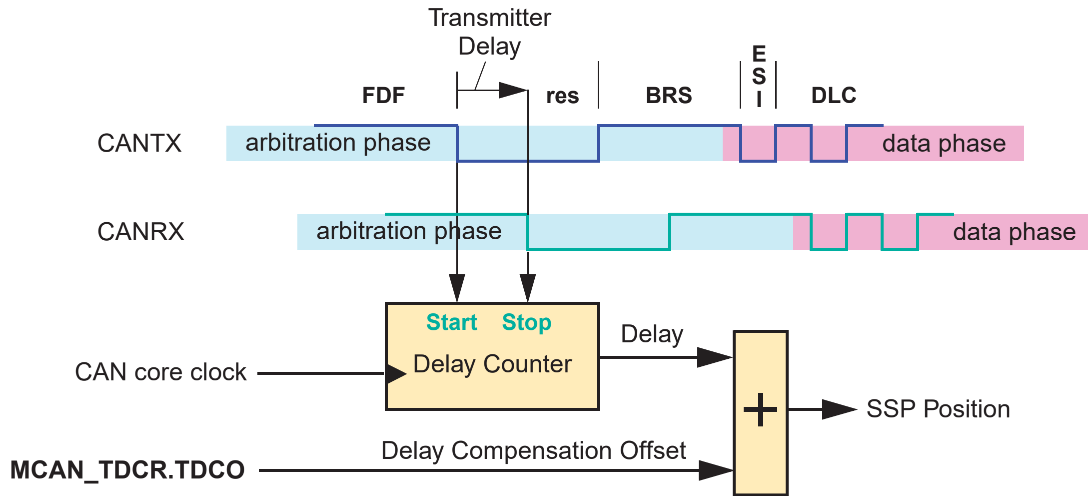
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less or equal 127 CAN core clock periods. In case this sum exceeds 127 CAN core clock periods, the maximum value of 127 CAN core clock periods is used for delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

66.5.1.4.2 Transmitter Delay Measurement

If transmitter delay compensation is enabled by programming MCAN_DBTP.TDC = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANRX of the transmitter.

The resolution of this measurement is one mtq.

Figure 66-2. Transmitter Delay Measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a to early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR.TDCF.

This defines a minimum value for the SSP position. Dominant edges on CANRX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF AND CANRX is low.

66.5.1.5 Restricted Operation Mode

In Restricted Operation mode, the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented. The processor can set the MCAN into Restricted Operation mode by setting bit MCAN_CCCR.ASM. The bit can only be set by the processor when both MCAN_CCCR.CCE and MCAN_CCCR.INIT are set to ‘1’. The bit can be reset by the processor at any time.

Restricted Operation mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

The Restricted Operation mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation mode after it has received a valid frame.

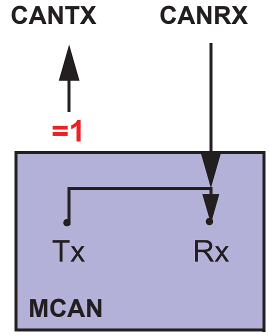
Note: The Restricted Operation Mode must not be combined with the Loop Back mode (internal or external).

66.5.1.6 Bus Monitoring Mode

The MCAN is set in Bus Monitoring mode by setting MCAN_CCCR.MON. In Bus Monitoring mode (see ISO11898-1, 10.12 Bus monitoring), the MCAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the MCAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring mode, the Tx Buffer Request Pending register (MCAN_TXBRP) is held in reset state.

The Bus Monitoring mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CANTX and CANRX to the MCAN in Bus Monitoring mode.

Figure 66-3. Pin Control in Bus Monitoring Mode



Bus Monitoring Mode

66.5.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the MCAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via MCAN_CCCR.DAR.

66.5.1.7.1 Frame Transmission in DAR Mode

In DAR mode, all transmissions are automatically cancelled after they start on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  – Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  – Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  – Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx set
  – Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  – Corresponding Tx Buffer Transmission Occurred bit MCAN_TXBTO.TOx not set
  – Corresponding Tx Buffer Cancellation Finished bit MCAN_TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = “10” (transmission in spite of cancellation).

66.5.1.8 Power-down (Sleep Mode)

The MCAN can be set into Power-down mode via bit MCAN_CCCR.CSR.

When all pending transmission requests have completed, the MCAN waits until bus idle state is detected. Then the MCAN sets MCAN_CCCR.INIT to prevent any further CAN transfers. Now the MCAN acknowledges that it is ready for power down by setting to one the bit MCAN_CCCR.CSA. In this state, before the clocks are switched off, further register accesses can be made. A write access to MCAN_CCCR.INIT will have no effect. Now the bus clock (peripheral clock) and the CAN core clock may be switched off.

To leave Power-down mode, the application has to turn on the MCAN clocks before clearing CC Control Register flag MCAN_CCCR.CSR. The MCAN will acknowledge this by clearing MCAN_CCCR.CSA. The application can then restart CAN communication by clearing the bit CCCR.INIT.

66.5.1.9 Test Modes

To enable write access to the MCAN Test register (MCAN_TEST) (see Section 7.6), bit MCAN_CCCR.TEST must be set. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANTX by programming MCAN_TEST.TX. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin CANRX can be read from MCAN_TEST.RX. Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between CAN clock and system bus clock domain, there may be a delay of several system bus clock periods between writing to MCAN_TEST.TX until the new configuration is visible at output pin CANTX. This applies also when reading input pin CANRX via MCAN_TEST.RX.

Note: Test modes should be used for production tests or self-test only. The software control for pin CANTX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

66.5.1.9.1 External Loop Back Mode

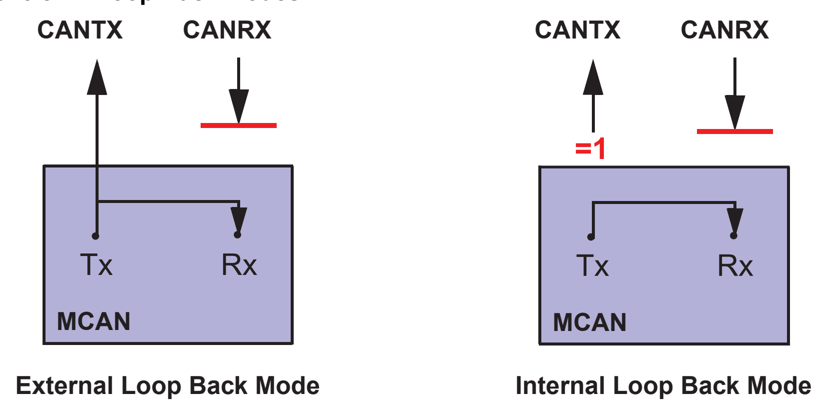
The MCAN can be set in External Loop Back mode by setting the bit MCAN_TEST.LBCK. In Loop Back mode, the MCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CANTX and CANRX to the MCAN in External Loop Back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back mode. In this mode, the MCAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANRX input pin is disregarded by the MCAN. The transmitted messages can be monitored at the CANTX pin.

66.5.1.9.2 Internal Loop Back Mode

Internal Loop Back mode is entered by setting bits MCAN_TEST.LBCK and MCAN_CCCR.MON. This mode can be used for a “Hot Selftest”, meaning the MCAN can be tested without affecting a running CAN system connected to the pins CANTX and CANRX. In this mode, pin CANRX is disconnected from the MCAN, and pin CANTX is held recessive. The figure below shows the connection of CANTX and CANRX to the MCAN when Internal Loop Back mode is enabled.

Figure 66-4. Pin Control in Loop Back Modes



66.5.2 Timestamp Generation

For timestamp generation, the MCAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP can be configured to clock the counter in multiples of CAN bit times (1…16). The counter is readable via MCAN_TSCV.TSC. A write access to the Timestamp Counter Value register (MCAN_TSCV) resets the counter to zero. When the timestamp counter wraps around, the interrupt flag MCAN_IR.TSW is set.

On start of frame reception / transmission, the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

By programming bit MCAN_TSCC.TSS, an external 16-bit timestamp can be used. See Timestamping for details.

66.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO, the MCAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP as the Timestamp Counter. The Timeout Counter is configured via the Timeout Counter Configuration register (MCAN_TOCC). The actual counter value can be read from MCAN_TOCV.TOC. The Timeout Counter can only be started while MCAN_CCCR.INIT = ‘0’. It is stopped when MCAN_CCCR.INIT = ‘1’, e.g. when the MCAN enters Bus_Off state.

The operating mode is selected by MCAN_TOCC.TOS. When operating in Continuous mode, the counter starts when MCAN_CCCR.INIT is reset. A write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored. Writing to MCAN_TOCV has no effect.

When the counter reaches zero, interrupt flag MCAN_IR.TOO is set. In Continuous mode, the counter is immediately restarted at MCAN_TOCC.TOP.

Note: The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the bit rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

66.5.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

66.5.4.1 Acceptance Filtering

The MCAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC)
- Standard ID Filter Configuration (MCAN_SIDFC)
- Extended ID Filter Configuration (MCAN_XIDFC)
- Extended ID and Mask (MCAN_XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag (MCAN_IR.HPM)
- Set High Priority Message interrupt flag (MCAN_IR.HPM) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the effected Rx Buffer or Rx FIFO:

- Rx Buffer
  New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC.

- Rx FIFO
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC. In case the matching Rx FIFO is operated in Overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

66.5.4.1.1 Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

There are two possibilities when range filtering is used together with extended frames:

- EFT = “00”: The Message ID of received frames is ANDed with MCAN_XIDAM before the range filter is applied.
- EFT = “11”: MCAN_XIDAM is not used for range filtering.

66.5.4.1.2 Filter for Specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

66.5.4.1.3 Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

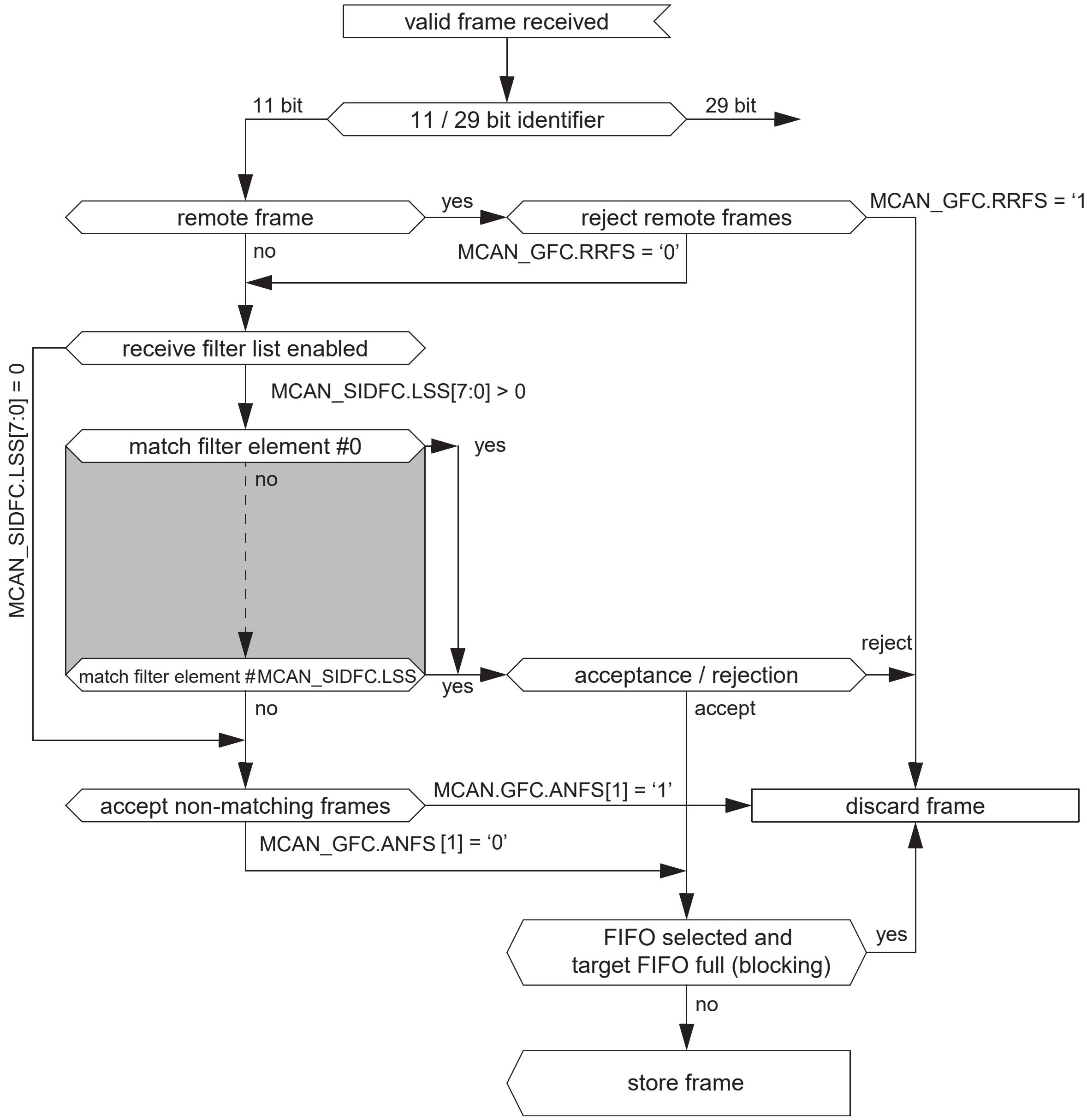
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

66.5.4.1.4 Standard Message ID Filtering

The figure below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Standard Message ID Filter Element.

Controlled by MCAN_GFC and MCAN_SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 66-5. Standard Message ID Filter Path



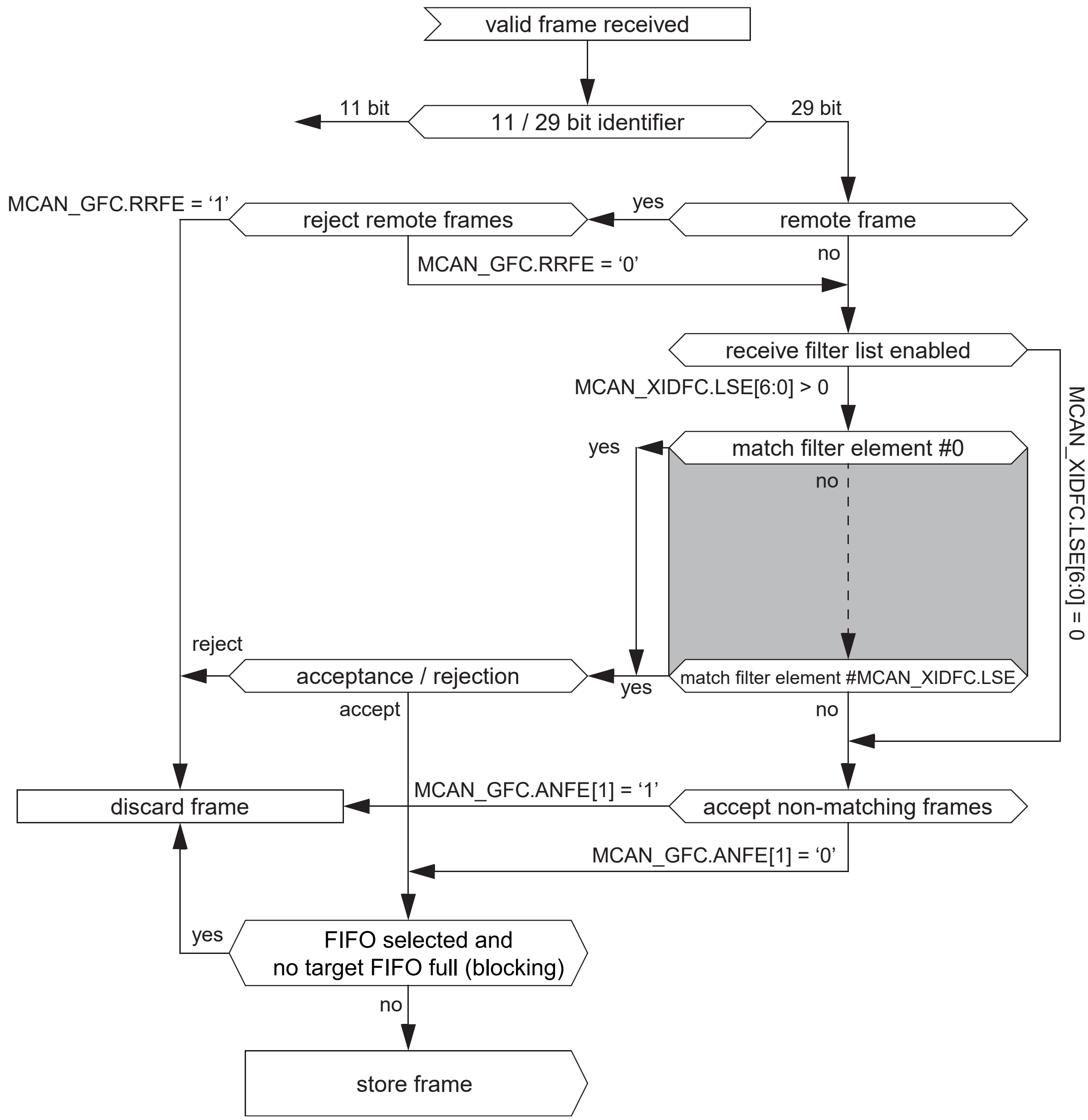
Extended Message ID Filtering

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in Extended Message ID Filter Element.

Controlled by MCAN_GFC and MCAN_XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

MCAN_XIDAM is ANDed with the received identifier before the filter list is executed.

Figure 66-6. Extended Message ID Filter Path



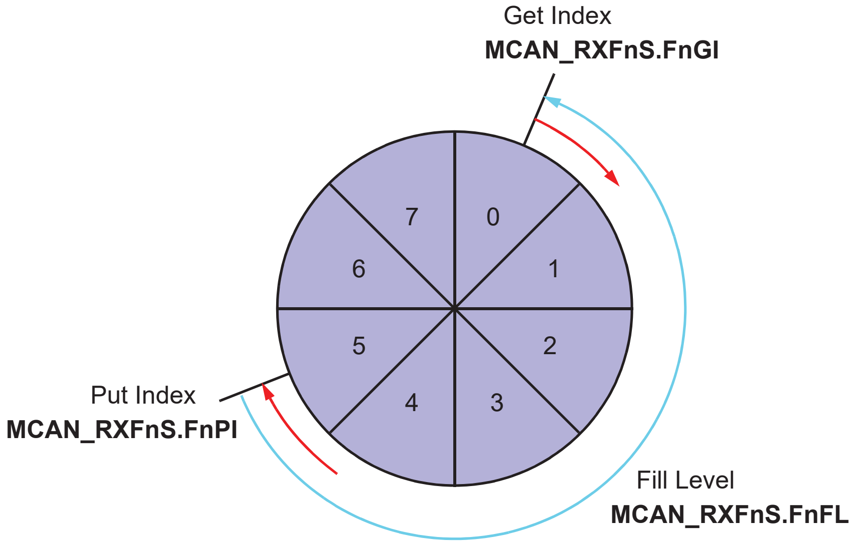
66.5.4.2 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via the Rx FIFO 0 Configuration register (MCAN_RXF0C) and the Rx FIFO 1 Configuration register (MCAN_RXF1C).

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1, see Acceptance Filtering. The Rx FIFO element is described in Rx Buffer and FIFO Element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by MCAN_RXFnC.FnWM, interrupt flag MCAN_IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index, an Rx FIFO Full condition is signalled by MCAN_RXFnS.FnF. In addition, the interrupt flag MCAN_IR.RFnF is set.

Figure 66-7. Rx FIFO Status



When reading from an Rx FIFO, Rx FIFO Get Index MCAN_RXFnS.FnGI × FIFO Element Size has to be added to the corresponding Rx FIFO start address MCAN_RXFnC.FnSA.

Table 66-2. Rx Buffer / FIFO Element Size

MCAN_RXESC.RBDS[2:0] MCAN_RXESC.FnDS[2:0]	**Data Field** [bytes]	**FIFO Element Size** [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

66.5.4.2.1 Rx FIFO Blocking Mode

The Rx FIFO Blocking mode is configured by MCAN_RXFnC.FnOM = ‘0’. This is the default operating mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by MCAN_RXFnS.FnF = ‘1’. In addition, the interrupt flag MCAN_IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by MCAN_RXFnS.RFnL = ‘1’. In addition, the interrupt flag MCAN_IR.RFnL is set.

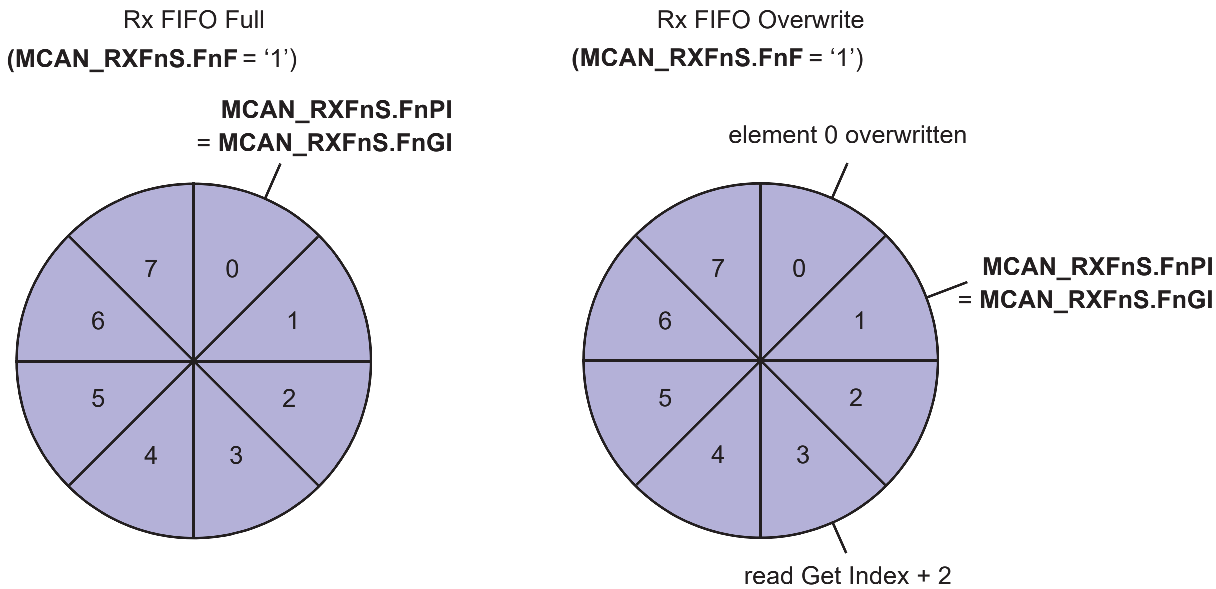
66.5.4.2.2 Rx FIFO Overwrite Mode

The Rx FIFO Overwrite mode is configured by MCAN_RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (MCAN_RXFnS.FnPI = MCAN_RXFnS.FnGI) is signalled by MCAN_RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in Overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the processor is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the processor accesses the Rx FIFO. The figure below shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 66-8. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS.FnF = '0').

66.5.4.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via MCAN_RXBC.RBSA.

For each Rx Buffer, a Standard or Extended Message ID Filter Element with SFEC / EFEC = 7 and SFID2 / EFID2[10:9] = 0 has to be configured.

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition, the flag MCAN_IR.DRX (Message stored in dedicated Rx Buffer) in MCAN_IR is set.

Table 66-3. Example Filter Configuration for Rx Buffers

Filter Element	**SFID1[10:0]** EFID1[28:0]	**SFID2[10:9]** EFID2[10:9]	**SFID2[5:0]** EFID2[5:0]
0	ID message 1	0	0
1	ID message 2	0	1
2	ID message 3	0	2

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in the New Data 1 register (MCAN_NDAT1) and New Data 2 register (MCAN_NDAT2) is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the processor by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

66.5.4.3.1 Rx Buffer Handling

- Reset interrupt flag IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

66.5.4.4 Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see Rx Buffer and FIFO Element).

Advantage: Fixed start address for the DMA transfers (relative to MCAN_RXBC.RBSA), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = ‘111’ have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output m_can_dma_req is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the MCAN while m_can_dma_req is activated. The behavior is similar to that of an Rx Buffer with its New Data flag set.

After the DMA has completed, the MCAN is prepared to receive the next set of debug messages.

66.5.4.4.1 Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to “111”. In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning. While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor MCAN_IR.DRX are set. The reception of debug messages can be monitored via RXF1S.DMS.

Table 66-4. Example Filter Configuration for Debug Messages

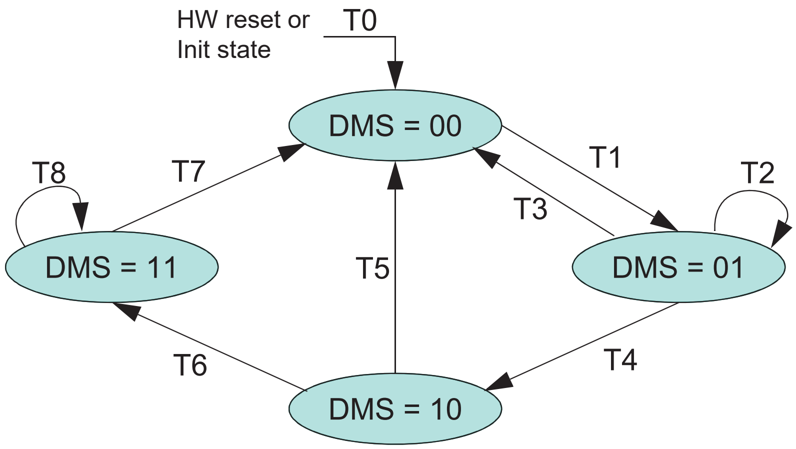
Filter Element	**SFID1[10:0]** EFID1[28:0]	**SFID2[10:9]** EFID2[10:9]	**SFID2[5:0]** EFID2[5:0]
0	ID debug message A	1	11 1101
1	ID debug message B	2	11 1110
2	ID debug message C	3	11 1111

66.5.4.4.2 Debug Message Handling

The debug message handling state machine ensures that debug messages are stored to three consecutive Rx Buffers in the correct order. If some messages are missing, the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in the correct order.

The status of the debug message handling state machine is signalled via MCAN_RXF1S.DMS.

Figure 66-9. Debug Message Handling State Machine



T0: reset m_can_dma_req output, enable reception of debug messages A, B, and C

T1: reception of debug message A

T2: reception of debug message A

T3: reception of debug message C

T4: reception of debug message B

T5: reception of debug messages A, B

T6: reception of debug message C

T7: DMA transfer completed

T8: reception of debug message A,B,C (message rejected)

66.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in Tx Buffer Element. The following table describes the possible configurations for frame transmission.

Table 66-5. Possible Configurations for Frame Transmission

MCAN_CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when MCAN_TXBRP is updated, or when a transmission has been started.

66.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit MCAN_CCCR.TXP. If the bit is set, the MCAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (MCAN_CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

66.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the processor. Each dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first. These Tx Buffers shall be requested in ascending order, with the lowest buffer number first. Alternatively, all Tx Buffers configured with the same Message ID can be requested simultaneously by a single write access to TXBAR.

If the data section has been updated, a transmission is requested by an Add Request via MCAN_TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see the table below). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0…31) × Element Size to the Tx Buffer Start Address TXBC.TBSA.

Table 66-6. Tx Buffer/FIFO/Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
0	8	4
1	12	5
2	16	6
3	20	7
4	24	8
5	32	10
6	48	14
7	64	18

66.5.5.3 Tx FIFO

Tx FIFO operation is configured by programming MCAN_TXBC.TFQM to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The MCAN calculates the Tx FIFO Free Level MCAN_TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (MCAN_TXFQS.TFQF = '1') is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via MCAN_TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see the table Tx Buffer/FIFO/ Queue Element Size). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

66.5.5.4 Tx Queue

Tx Queue operation is configured by programming MCAN_TXBC.TFQM to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Tx Queue buffers are configured with the same Message ID, the transmission order depends on numbers of buffers where messages were stored for transmission. As these buffer numbers depend on the then current states of the Put index, a prediction of the transmission order is not possible

New messages have to be written to the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. The Put index always points to the free buffer of the Tx Queue with the lowest buffer number. In case that the Tx Queue is full (MCAN_TXFQS.TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

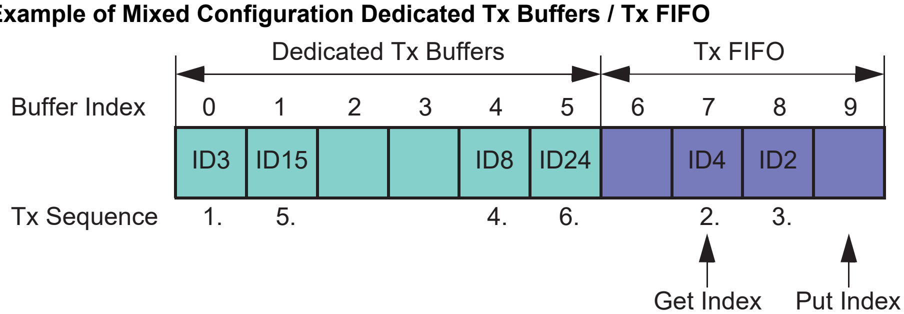
The application may use register MCAN_TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see the table Tx Buffer/FIFO/ Queue Element Size). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

66.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 66-10. Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO



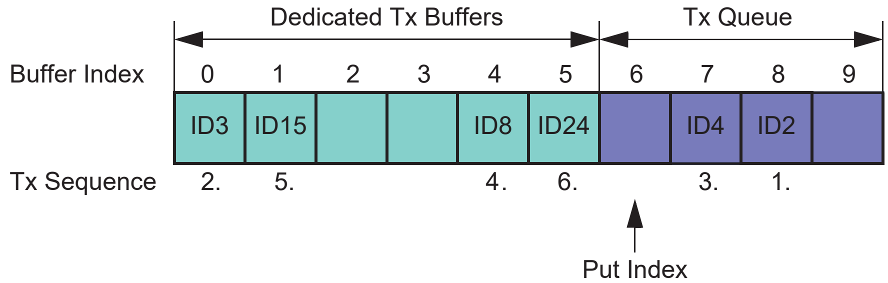
Tx prioritization:

- Scan dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by MCAN_TXFS.TFGI)
- Buffer with lowest Message ID gets highest priority and is transmitted next

66.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Queue Buffers is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 66-11. Example of Mixed Configuration Dedicated Tx Buffers / Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

66.5.5.7 Transmit Cancellation

The MCAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR-based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer, the processor has to write a '1' to the corresponding bit position (=number of Tx Buffer) of register MCAN_TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of register MCAN_TXBCF to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO and MCAN_TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding MCAN_TXBCF bit is set.

Note: In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

66.5.5.8 Tx Event Handling

To support Tx event handling the MCAN has implemented a Tx Event FIFO. After the MCAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Debug on CAN Support.

When a Tx Event FIFO full condition is signalled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag MCAN_IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by MCAN_TXEFC.EFWM, interrupt flag MCAN_IR.TEFW is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA.

66.5.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index in the registers MCAN_RXF0A, MCAN_RXF1A and MCAN_TXEFA. Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the processor has free access to the MCAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN does not check for erroneous values.

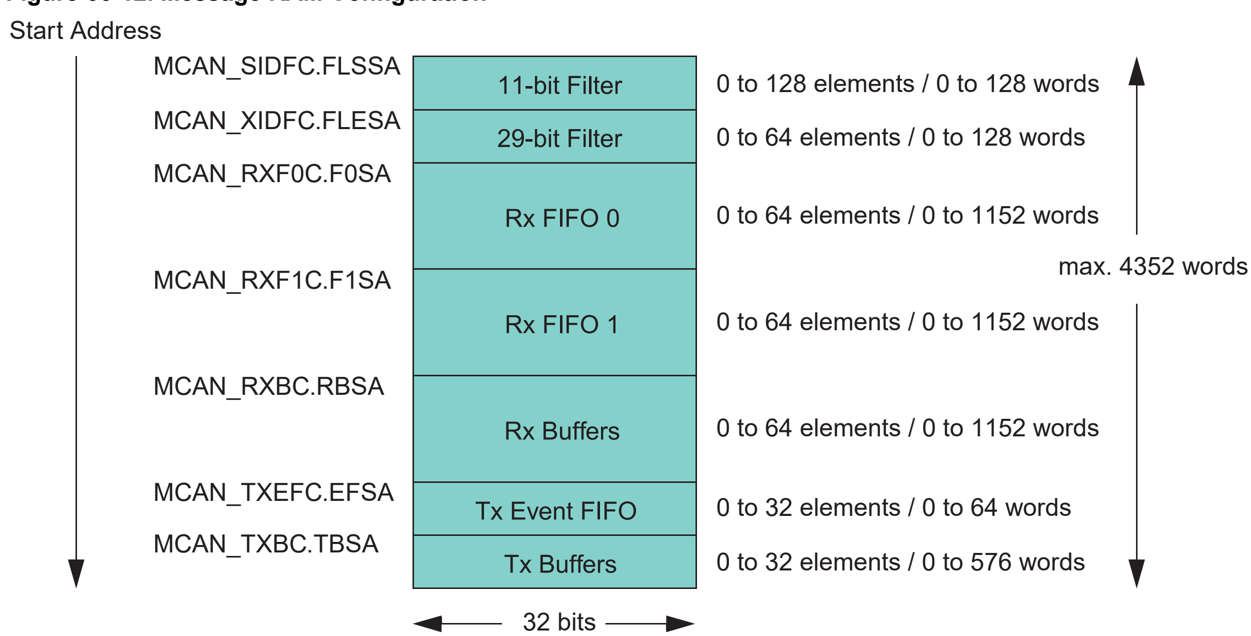
66.5.7 Message RAM

66.5.7.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The MCAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode, the required Message RAM size depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via MCAN_RXESC.F0DS, MCAN_RXESC.F1DS, MCAN_RXESC.RBDS, and MCAN_TXESC.TBDS.

Figure 66-12. Message RAM Configuration



When the MCAN addresses the Message RAM, it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses; i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Note: The MCAN does not check for erroneous configuration of the Message RAM. The configuration of the start addresses of the different sections and the number of elements of each section must be checked carefully to avoid falsification or loss of data.

66.5.7.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in the table below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register MCAN_RXESC.

Table 66-7. Rx Buffer and FIFO Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>R0</td><td>ESI</td><td>XTD</td><td>RTR</td><td colspan="29">ID[28:0]</td></tr>
<tr><td>R1</td><td>ANMF</td><td colspan="7">FIDX[6:0]</td><td colspan="2">–</td><td>FDF</td><td>BRS</td><td colspan="4">DLC[3:0]</td><td colspan="16">RXTS[15:0]</td></tr>
<tr><td>R2</td><td colspan="8">DB3[7:0]</td><td colspan="8">DB2[7:0]</td><td colspan="8">DB1[7:0]</td><td colspan="8">DB0[7:0]</td></tr>
<tr><td>R3</td><td colspan="8">DB7[7:0]</td><td colspan="8">DB6[7:0]</td><td colspan="8">DB5[7:0]</td><td colspan="8">DB4[7:0]</td></tr>
<tr><td>...</td><td colspan="8">..</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>Rn</td><td colspan="8">DBm[7:0]</td><td colspan="8">DBm-1[7:0]DBM[7:0]</td><td colspan="8">DBm-2[7:0]DBM[7:0]</td><td colspan="8">DBm-3[7:0]DBM[7:0]</td></tr>
</table>

• **R0 Bit 31 ESI:** Error State Indicator

0: Transmitting node is error active.

1: Transmitting node is error passive.

• **R0 Bit 30 XTD:** Extended Identifier

Signals to the processor whether the received frame has a standard or extended identifier.

0: 11-bit standard identifier.

1: 29-bit extended identifier.

• **R0 Bit 29 RTR:** Remote Transmission Request

Signals to the processor whether the received frame is a data frame or a remote frame.

0: Received frame is a data frame.

1: Received frame is a remote frame.

Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), bit RTR reflects the state of the reserved bit r1.

• **R0 Bits 28:0 ID[28:0]:** Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

• **R1 Bit 31 ANMF:** Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via MCAN_GFC.ANFS and MCAN_GFC.ANFE.

0: Received frame matching filter index FIDX.

1: Received frame did not match any Rx filter element.

• **R1 Bits 30:24 FIDX[6:0]:** Filter Index

0-127: Index of matching Rx acceptance filter element (invalid if ANMF = ‘1’).
Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

• **R1 Bit 21 FDF:** FD Format

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

• **R1 Bit 20 BRS:** Bit Rate Switch

0: Frame received without bit rate switching.

1: Frame received with bit rate switching.

Note:
Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

• **R1 Bits 19:16 DLC[3:0]:** Data Length Code

0-8: CAN + CAN FD: received frame has 0-8 data bytes.

9-15: CAN: received frame has 8 data bytes.

9-15: CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

• **R1 Bits 15:0 RXTS[15:0]:** Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.

• **R2 Bits 31:24 DB3[7:0]:** Data Byte 3

• **R2 Bits 23:16 DB2[7:0]:** Data Byte 2

• **R2 Bits 15:8 DB1[7:0]:** Data Byte 1

• **R2 Bits 7:0 DB0[7:0]:** Data Byte 0

• **R3 Bits 31:24 DB7[7:0]:** Data Byte 7

• **R3 Bits 23:16 DB6[7:0]:** Data Byte 6

• **R3 Bits 15:8 DB5[7:0]:** Data Byte 5

• **R3 Bits 7:0 DB4[7:0]:** Data Byte 4

... ...

• **Rn Bits 31:24 DBm[7:0]:** Data Byte m

• **Rn Bits 23:16 DBm-1[7:0]:** Data Byte m-1

• **Rn Bits 15:8 DBm-2[7:0]:** Data Byte m-2

• **Rn Bits 7:0 DBm-3[7:0]:** Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3 ..17) are used for storage of a CAN message's data field.

66.5.7.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue.
The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 66-8. Tx Buffer Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>T0</td><td>ESI</td><td>XTD</td><td>RTR</td><td colspan="29">ID[28:0]</td></tr>
<tr><td>T1</td><td colspan="8">MM[7:0]</td><td>EFC</td><td>reserved</td><td>FDF</td><td>BRS</td><td colspan="4">DLC[3:0]</td><td colspan="16">reserved</td></tr>
<tr><td>T2</td><td colspan="8">DB3[7:0]</td><td colspan="8">DB2[7:0]</td><td colspan="8">DB1[7:0]</td><td colspan="8">DB0[7:0]</td></tr>
<tr><td>T3</td><td colspan="8">DB7[7:0]</td><td colspan="8">DB6[7:0]</td><td colspan="8">DB5[7:0]</td><td colspan="8">DB4[7:0]</td></tr>
<tr><td>...</td><td colspan="8">..</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>Tn</td><td colspan="8">DBm[7:0]</td><td colspan="8">DBm-1[7:0]DBM[7:0]</td><td colspan="8">DBm-2[7:0]DBM[7:0]</td><td colspan="8">DBm-3[7:0]DBM[7:0]</td></tr>
</table>

• **T0 Bit 30 ESI:** Error State Indicator

0: ESI bit in CAN FD format depends only on error passive flag

1: ESI bit in CAN FD format transmitted recessive

Note: The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive. This feature can be used in gateway applications when a message from an error passive node is routed to another CAN network.

• **T0 Bit 30 XTD:** Extended Identifier

0: 11-bit standard identifier.

1: 29-bit extended identifier.

• **T0 Bit 29 RTR:** Remote Transmission Request

0: Transmit data frame.

1: Transmit remote frame.

Note: When RTR = 1, the MCAN transmits a remote frame according to ISO11898-1, even if MCAN_CCCR.FDOE enables the transmission in CAN FD format.

• **T0 Bits 28:0 ID[28:0]:** Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

• **T1 Bits 31:24 MM[7:0]:** Message Marker

Written by processor during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

• **T1 Bit 23 EFC:** Event FIFO Control

0: Do not store Tx events.

1: Store Tx events.

• **T1 Bit 21 FDF:** FD Format

0: Frame transmitted in Classic CAN format

1: Frame transmitted in CAN FD format

• **T1 Bit 20 BRS:** Bit Rate Switching

0: CAN FD frames transmitted without bit rate switching

1: CAN FD frames transmitted with bit rate switching

Note:
Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled (MCAN_CCCR.FDOE = 1). Bit BRS is only evaluated when in addition MCAN_CCCR.BRSE = 1.

• **T1 Bits 19:16 DLC[3:0]:** Data Length Code

0-8: CAN + CAN FD: transmit frame has 0-8 data bytes.

9-15: CAN: transmit frame has 8 data bytes.

9-15: CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes.

• **T2 Bits 31:24 DB3[7:0]:** Data Byte 3

• **T2 Bits 23:16 DB2[7:0]:** Data Byte 2

• **T2 Bits 15:8 DB1[7:0]:** Data Byte 1

• **T2 Bits 7:0 DB0[7:0]:** Data Byte 0

• **T3 Bits 31:24 DB7[7:0]:** Data Byte 7

• **T3 Bits 23:16 DB6[7:0]:** Data Byte 6

• **T3 Bits 15:8 DB5[7:0]:** Data Byte 5

• **T3 Bits 7:0 DB4[7:0]:** Data Byte 4

... ...

• **Tn Bits 31:24 DBm[7:0]:** Data Byte m

• **Tn Bits 23:16 DBm-1[7:0]:** Data Byte m-1

• **Tn Bits 15:8 DBm-2[7:0]:** Data Byte m-2

• **Tn Bits 7:0 DBm-3[7:0]:** Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3 ..17) are used for storage of a CAN message's data field.

66.5.7.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the processor gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 66-9. Tx Event FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTD	RTR	ID[28:0]																												
E1	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]				TXTS[15:0]															

• **E0 Bit 31 ESI:** Error State Indicator

0: Transmitting node is error active.

1: Transmitting node is error passive.

• **E0 Bit 30 XTD:** Extended Identifier

0: 11-bit standard identifier.

1: 29-bit extended identifier.

• **E0 Bit 29 RTR:** Remote Transmission Request

0: Data frame transmitted.

1: Remote frame transmitted.

• **E0 Bits 28:0 ID[28:0]:** Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

• **E1 Bits 31:24 MM[7:0]:** Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

• **E1 Bit 23:22 ET[1:0]:** Event Type

0: Reserved

1: Tx event

2: Transmission in spite of cancellation (always set for transmissions in DAR mode)

3: Reserved

• **E1 Bit 21 FDF:** FD Format

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

• **E1 Bit 20 BRS:** Bit Rate Switch

0: Frame transmitted without bit rate switching.

1: Frame transmitted with bit rate switching.

• **E1 Bits 19:16 DLC[3:0]:** Data Length Code

0-8: CAN + CAN FD: frame with 0-8 data bytes transmitted.

9-15: CAN: frame with 8 data bytes transmitted.

9-15: CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted

• **E1 Bits 15:0 TXTS[15:0]:** Tx Timestamp

Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.

66.5.7.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA plus the index of the filter element (0…127).

Table 66-10. Standard Message ID Filter Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT[1:0]		SFEC[2:0]			SFID[10:0]											–					SFID[10:0]										

• **Bits 31:30 SFT[1:0]:** Standard Filter Type

Value	Description
0	Range filter from SF1ID to SF2ID (SF2ID ≥ SF1ID)
1	Dual ID filter for SF1ID or SF2ID
2	Classic filter: SF1ID = filter, SF2ID = mask
3	Reserved

• **Bit 29:27 SFEC[2:0]:** Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored

• **Bits 26:16 SFID1[10:0]:** Standard Filter ID 1

First ID of standard ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.

• **Bits 10:0 SFID2[10:0]:** Standard Filter ID 2

This field has a different meaning depending on the configuration of SFEC:

- SFEC = “001”...“110”–Second ID of standard ID filter element
- SFEC = “111”–Filter for Rx Buffers or for debug messages

SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in a Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

SFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

66.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 66-11. Extended Message ID Filter Elemen

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0	EFEC[2:0]			EFID[28:0]																												
F1	EFT[1:0]		–	EFID[28:0]																												

- **F0 Bit 31:29 EFEC[2:0]:** Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

Value	Description
0	Disable filter element
1	Store in Rx FIFO 0 if filter matches
2	Store in Rx FIFO 1 if filter matches
3	Reject ID if filter matches
4	Set priority if filter matches
5	Set priority and store in FIFO 0 if filter matches
6	Set priority and store in FIFO 1 if filter matches
7	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

- **F0 Bits 28:0 EFID1[28:0]:** Extended Filter ID 1

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

- **F1 Bits 31:30 EFT[1:0]:** Extended Filter Type

Value	Description
0	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)
1	Dual ID filter for EF1ID or EF2ID
2	Classic filter: EF1ID = filter, EF2ID = mask
3	Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

• **F1 Bits 28:0 EFID2[28:0]:** Extended Filter ID 2

This field has a different meaning depending on the configuration of EFEC:

- EFEC = “001”...“110”–Second ID of extended ID filter element
- EFEC = “111”–Filter for Rx Buffers or for debug messages

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

Value	Description
0	Store message in an Rx buffer
1	Debug Message A
2	Debug Message B
3	Debug Message C

EFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

66.5.8 Hardware Reset Description

After hardware reset, the registers of the MCAN hold the reset values listed in the register descriptions. Additionally the Bus_Off state is reset and the output CANTX is set to recessive (HIGH). The value 0x0001 (MCAN_CCCR.INIT = ‘1’) in the CC Control register enables software initialization. The MCAN does not influence the CAN bus until the processor resets MCAN_CCCR.INIT to ‘0’.

66.5.9 Access to Reserved Register Addresses

In case the application software accesses one of the reserved addresses in the MCAN register map (read or write access), interrupt flag MCAN_IR.ARA is set and, if enabled, the selected interrupt line is risen.

66.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	MCAN_ENDN	31:24	ETV[31:24]							
		23:16	ETV[23:16]							
		15:8	ETV[15:8]							
		7:0	ETV[7:0]							
0x08 ... 0x0B	Reserved									
0x0C	MCAN_DBTP	31:24								
		23:16	TDC			DBRP[4:0]				
		15:8				DTSEG1[4:0]				
		7:0	DTSEG2[3:0]					DSJW[2:0]		
0x10	MCAN_TEST	31:24								
		23:16								
		15:8								
		7:0	RX	TX[1:0]		LBCK				
0x14	MCAN_RWD	31:24								
		23:16								
		15:8	WDV[7:0]							
		7:0	WDC[7:0]							
0x18	MCAN_CCCR	31:24								
		23:16								
		15:8	NISO	TXP	EFBI	PXHD			BRSE	FDOE
		7:0	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
0x1C	MCAN_NBTP	31:24	NSJW[6:0]							NBRP[8]
		23:16	NBRP[7:0]							
		15:8	NTSEG1[7:0]							
		7:0		NTSEG2[6:0]						
0x20	MCAN_TSCC	31:24								
		23:16					TCP[3:0]			
		15:8								
		7:0							TSS[1:0]	
0x24	MCAN_TSCV	31:24								
		23:16								
		15:8	TSC[15:8]							
		7:0	TSC[7:0]							
0x28	MCAN_TOCC	31:24	TOP[15:8]							
		23:16	TOP[7:0]							
		15:8								
		7:0						TOS[1:0]		ETOC
0x2C	MCAN_TOCV	31:24								
		23:16								
		15:8	TOC[15:8]							
		7:0	TOC[7:0]							
0x30 ... 0x3F	Reserved									
0x40	MCAN_ECR	31:24								
		23:16	CEL[7:0]							
		15:8	RP	REC[6:0]						
		7:0	TEC[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	MCAN_PSR	31:24								
		23:16		TDCV[6:0]						
		15:8		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
		7:0	BO	EW	EP	ACT[1:0]		LEC[2:0]		
0x48	MCAN_TDCR	31:24								
		23:16								
		15:8		TDCO[6:0]						
		7:0		TDCF[6:0]						
0x4C ... 0x4F	Reserved									
0x50	MCAN_IR	31:24			ARA	PED	PEA	WDI	BO	EW
		23:16	EP	ELO			DRX	TOO	MRAF	TSW
		15:8	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
		7:0	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
0x54	MCAN_IE	31:24			ARAE	PEDE	PEAE	WDIE	BOE	EWE
		23:16	EPE	ELOE			DRXE	TOOE	MRAFE	TSWE
		15:8	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
		7:0	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
0x58	MCAN_ILS	31:24			ARAL	PEDL	PEAL	WDIL	BOL	EWL
		23:16	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
		15:8	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
		7:0	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
0x5C	MCAN_ILE	31:24								
		23:16								
		15:8								
		7:0							EINT1	EINT0
0x60 ... 0x7F	Reserved									
0x80	MCAN_GFC	31:24								
		23:16								
		15:8								
		7:0			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
0x84	MCAN_SIDFC	31:24								
		23:16	LSS[7:0]							
		15:8	FLSSA[13:6]							
		7:0	FLSSA[5:0]							
0x88	MCAN_XIDFC	31:24								
		23:16		LSE[6:0]						
		15:8	FLESA[13:6]							
		7:0	FLESA[5:0]							
0x8C ... 0x8F	Reserved									
0x90	MCAN_XIDAM	31:24				EIDM[28:24]				
		23:16	EIDM[23:16]							
		15:8	EIDM[15:8]							
		7:0	EIDM[7:0]							
0x94	MCAN_HPMS	31:24								
		23:16								
		15:8	FLST	FIDX[6:0]						
		7:0	MSI[1:0]		BIDX[5:0]					
0x98	MCAN_NDAT1	31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
		7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x9C	MCAN_NDAT2	31:24	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
		23:16	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
		15:8	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
		7:0	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
0xA0	MCAN_RXF0C	31:24	F0OM	F0WM[6:0]						
		23:16		F0S[6:0]						
		15:8	F0SA[13:6]							
		7:0	F0SA[5:0]							
0xA4	MCAN_RXF0S	31:24							RF0L	F0F
		23:16			F0PI[5:0]					
		15:8			F0GI[5:0]					
		7:0		F0FL[6:0]						
0xA8	MCAN_RXF0A	31:24								
		23:16								
		15:8								
		7:0			F0AI[5:0]					
0xAC	MCAN_RXBC	31:24								
		23:16								
		15:8	RBSA[13:6]							
		7:0	RBSA[5:0]							
0xB0	MCAN_RXF1C	31:24	F1OM	F1WM[6:0]						
		23:16		F1S[6:0]						
		15:8	F1SA[13:6]							
		7:0	F1SA[5:0]							
0xB4	MCAN_RXF1S	31:24	DMS[1:0]						RF1L	F1F
		23:16			F1PI[5:0]					
		15:8			F1GI[5:0]					
		7:0		F1FL[6:0]						
0xB8	MCAN_RXF1A	31:24								
		23:16								
		15:8								
		7:0			F1AI[5:0]					
0xBC	MCAN_RXESC	31:24								
		23:16								
		15:8						RBDS[2:0]		
		7:0		F1DS[2:0]				F0DS[2:0]		
0xC0	MCAN_TXBC	31:24		TFQM	TFQS[5:0]					
		23:16			NDTB[5:0]					
		15:8	TBSA[13:6]							
		7:0	TBSA[5:0]							
0xC4	MCAN_TXFQS	31:24								
		23:16			TFQF	TFQPI[4:0]				
		15:8				TFGI[4:0]				
		7:0			TFFL[5:0]					
0xC8	MCAN_TXESC	31:24								
		23:16								
		15:8								
		7:0						TBDS[2:0]		
0xCC	MCAN_TXBRP	31:24	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
		23:16	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
		15:8	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
		7:0	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
0xD0	MCAN_TXBAR	31:24	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
		23:16	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
		15:8	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
		7:0	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xD4	MCAN_TXBCR	31:24	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
		23:16	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
		15:8	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
		7:0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
0xD8	MCAN_TXBTO	31:24	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
		23:16	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
		15:8	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
		7:0	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
0xDC	MCAN_TXBCF	31:24	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
		23:16	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
		15:8	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
		7:0	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
0xE0	MCAN_TXBTIE	31:24	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
		23:16	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
		15:8	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
		7:0	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
0xE4	MCAN_TXBCIE	31:24	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
		23:16	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
		15:8	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
		7:0	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
0xE8 ... 0xEF	Reserved									
0xF0	MCAN_TXEFC	31:24			EFWM[5:0]					
		23:16			EFS[5:0]					
		15:8	EFSA[13:6]							
		7:0	EFSA[5:0]							
0xF4	MCAN_TXEFS	31:24							TEFL	EFF
		23:16				EFPI[4:0]				
		15:8				EFGI[4:0]				
		7:0			EFFL[5:0]					
0xF8	MCAN_TXEFA	31:24								
		23:16								
		15:8								
		7:0				EFAI[4:0]				

66.6.1 MCAN Endian Register

Name: MCAN_ENDN
Offset: 0x04
Reset: 0x87654321
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ETV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	ETV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8
	ETV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	0	0	0	1	1

Bit	7	6	5	4	3	2	1	0
	ETV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	0	0	0	0	1

Bits 31:0 – ETV[31:0] Endianness Test Value
The endianness test value is 0x87654321.

66.6.2 MCAN Data Bit Timing and Prescaler Register

Name: MCAN_DBTP
Offset: 0x0C
Reset: 0x00000A33
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

The CAN bit time may be programmed in the range of 4 to 25 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 CAN core clock periods. t_q = (DBRP + 1) CAN core clock periods.

DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [DTSEG1 + DTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

With a CAN core clock frequency of 8 MHz, the reset value of 0x00000A33 configures the MCAN for a fast bit rate of 500 kbit/s.

The bit rate configured for the CAN FD data phase via MCAN_DBTP must be higher than or equal to the bit rate configured for the arbitration phase via MCAN_NBTP.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TDC			DBRP[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DTSEG1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	DTSEG2[3:0]					DSJW[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	1	1		0	1	1

Bit 23 – TDC Transmitter Delay Compensation
0 (DISABLED): Transmitter Delay Compensation disabled.
1 (ENABLED): Transmitter Delay Compensation enabled.

Bits 20:16 – DBRP[4:0] Data Bit Rate Prescaler
The value by which the peripheral clock is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 12:8 – DTSEG1[4:0] Data Time Segment Before Sample Point
0: Forbidden.
1 to 31: The duration of time segment is t_q x (DTSEG1 + 1).

Bits 7:4 – DTSEG2[3:0] Data Time Segment After Sample Point
The duration of time segment is t_q x (DTSEG2 + 1).

Bits 2:0 – DSJW[2:0] Data (Re) Synchronization Jump Width
The duration of a synchronization jump is t_q x (DSJW + 1).

66.6.3 MCAN Test Register

Name: MCAN_TEST
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Write access to the Test Register has to be enabled by setting bit MCAN_CCCR.TEST to ‘1’.

All MCAN Test Register functions are set to their reset values when bit MCAN_CCCR.TEST is cleared.

Loop Back mode and software control of pin CANTX are hardware test modes. Programming of TX ≠ 0 disturbs the message transfer on the CAN bus.

The reset value for MCAN_TEST.RX is undefined.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RX	TX[1:0]		LBCK				
Access	R	R/W	R/W	R/W				
Reset	x	0	0	0				

Bit 7 – RX Receive Pin
Monitors the actual value of pin CANRX.
The reset value for this bit is undefined.

Value	Description
0	The CAN bus is dominant (CANRX = ‘0’).
1	The CAN bus is recessive (CANRX = ‘1’).

Bits 6:5 – TX[1:0] Control of Transmit Pin

Value	Name	Description
0	RESET	Reset value, CANTX controlled by the CAN Core, updated at the end of the CAN bit time.
1	SAMPLE_POINT_MONITORING	Sample Point can be monitored at pin CANTX.
2	DOMINANT	Dominant (‘0’) level at pin CANTX.
3	RECESSIVE	Recessive (‘1’) at pin CANTX.

Bit 4 – LBCK Loop Back Mode
0 (DISABLED): Reset value. Loop Back mode is disabled.
1 (ENABLED): Loop Back mode is enabled (see Test Modes).

66.6.4 MCAN RAM Watchdog Register

Name: MCAN_RWD
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

The RAM Watchdog monitors the Message RAM response time. A Message RAM access via the MCAN's Generic Host Interface starts the Message RAM Watchdog Counter with the value configured by MCAN_RWD.WDC. The counter is reloaded with MCAN_RWD.WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the system bus clock (peripheral clock).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WDV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WDC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – WDV[7:0] Watchdog Value (read-only)
Watchdog Counter Value for the current message located in RAM.

Bits 7:0 – WDC[7:0] Watchdog Configuration (read/write)
Start value of the Message RAM Watchdog Counter. The counter is disabled when WDC is cleared.

66.6.5 MCAN CC Control Register

Name: MCAN_CCCR
Offset: 0x18
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	NISO	TXP	EFBI	PXHD			BRSE	FDOE
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0
Bit	7	6	5	4	3	2	1	0
	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit 15 – NISO Non-ISO Operation
If this bit is set, the MCAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0.

Value	Description
0	CAN FD frame format according to ISO11898-1 (default).
1	CAN FD frame format according to Bosch CAN FD Specification V1.0.

Bit 14 – TXP Transmit Pause (write protection)
If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Tx Handling).

Value	Description
0	Transmit pause disabled.
1	Transmit pause enabled.

Bit 13 – EFBI Edge Filtering during Bus Integration (write protection)

Value	Description
0	Edge filtering is disabled.
1	Edge filtering is enabled. Two consecutive dominant tq required to detect an edge for hard synchronization.

Bit 12 – PXHD Protocol Exception Event Handling (write protection)

Value	Description
0	Protocol exception handling enabled.
1	Protocol exception handling disabled.

Bit 9 – BRSE Bit Rate Switching Enable (write protection)
0 (DISABLED): Bit rate switching for transmissions disabled.
1 (ENABLED): Bit rate switching for transmissions enabled.

Bit 8 – FDOE CAN FD Operation Enable (write protection)
0 (DISABLED): FD operation disabled.
1 (ENABLED): FD operation enabled.

Bit 7 – TEST Test Mode Enable (write protection against '1')
0 (DISABLED): Normal operation, MCAN_TEST register holds reset values.
1 (ENABLED): Test mode, write access to MCAN_TEST register enabled.

Bit 6 – DAR Disable Automatic Retransmission (write protection)
0 (AUTO_RETX): Automatic retransmission of messages not transmitted successfully enabled.
1 (NO_AUTO_RETX): Automatic retransmission disabled.

Bit 5 – MON Bus Monitoring Mode (write protection against '1')
0 (DISABLED): Bus Monitoring mode is disabled.
1 (ENABLED): Bus Monitoring mode is enabled.

Bit 4 – CSR Clock Stop Request
0 (NO_CLOCK_STOP): No clock stop is requested.
1 (CLOCK_STOP): Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.

Bit 3 – CSA Clock Stop Acknowledge

Value	Description
0	No clock stop acknowledged.
1	MCAN may be set in power down by stopping the peripheral clock and the CAN core clock.

Bit 2 – ASM Restricted Operation Mode (write protection against '1')
For a description of the Restricted Operation mode see Restricted Operation Mode.
0 (NORMAL): Normal CAN operation.
1 (RESTRICTED): Restricted Operation mode active.

Bit 1 – CCE Configuration Change Enable (write protection)
0 (PROTECTED): The processor has no write access to the protected configuration registers.
1 (CONFIGURABLE): The processor has write access to the protected configuration registers (while MCAN_CCCR.INIT = '1').

Bit 0 – INIT Initialization
Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to ensure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.
0 (DISABLED): Normal operation.
1 (ENABLED): Initialization is started.

66.6.6 MCAN Nominal Bit Timing and Prescaler Register

Name: MCAN_NBTP
Offset: 0x1C
Reset: 0x06000A03
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN_CCCR.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 CAN core clock periods. $t_q = t_{core\ clock}$ x (NBRP + 1).

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

With a CAN core clock frequency of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kbit/s.

Bit	31	30	29	28	27	26	25	24
	NSJW[6:0]							NBRP[8]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bit	23	22	21	20	19	18	17	16
	NBRP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NTSEG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
		NTSEG2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	1	1

Bits 31:25 – NSJW[6:0] Nominal (Re) Synchronization Jump Width
0 to 127: The duration of a synchronization jump is t_q x (NSJW + 1).

Bits 24:16 – NBRP[8:0] Nominal Bit Rate Prescaler
0 to 511: The value by which the oscillator frequency is divided for generating the CAN time quanta. The CAN time is built up from a multiple of this quanta. CAN time quantum (tq) = $t_{core\ clock}$ x (NBRP + 1)

Bits 15:8 – NTSEG1[7:0] Nominal Time Segment Before Sample Point

Value	Description
`0`	Reserved; do not use.
`1 to 255`	The duration of time segment is t_q x (NTSEG1 + 1).

Bits 6:0 – NTSEG2[6:0] Nominal Time Segment After Sample Point

Value	Description
`0`	Reserved; do not use.
`1 to 127`	The duration of time segment is t_q x (NTSEG2 + 1).

66.6.7 MCAN Timestamp Counter Configuration Register

Name: MCAN_TSCC
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

For a description of the Timestamp Counter see Timestamp Generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					TCP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							TSS[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 19:16 – TCP[3:0] Timestamp Counter Prescaler
Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 1:0 – TSS[1:0] Timestamp Select
With CAN FD, an external counter is required for timestamp generation (TSS = 2).

Value	Name	Description
0	ALWAYS_0	Timestamp counter value always 0x0000
1	TCP_INC	Timestamp counter value incremented according to TCP
2	EXT_TIMESTAMP	External timestamp counter value used
3	ALWAYS_0	Timestamp counter value always 0x0000

66.6.8 MCAN Timestamp Counter Value Register

Name: MCAN_TSCV
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TSC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TSC[15:0] Timestamp Counter (cleared on write)
The internal/external Timestamp Counter value is captured on start of frame (both Receive and Transmit). When MCAN_TSCC.TSS = 1, the Timestamp Counter is incremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. A wrap around sets interrupt flag MCAN_IR.TSW. Write access resets the counter to zero.
When MCAN_TSCC.TSS = 2, TSC reflects the external Timestamp Counter value. Thus a write access has no impact.
Note: A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV.

66.6.9 MCAN Timeout Counter Configuration Register

Name: MCAN_TOCC
Offset: 0x28
Reset: 0xFFFF0000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

For a description of the Timeout Counter, see Timeout Counter.

Bit	31	30	29	28	27	26	25	24
	TOP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	TOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TOS[1:0]		ETOC
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:16 – TOP[15:0] Timeout Period
Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits 2:1 – TOS[1:0] Timeout Select
When operating in Continuous mode, a write to MCAN_TOCV presets the counter to the value configured by MCAN_TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by MCAN_TOCC.TOP. Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0	CONTINUOUS	Continuous operation.
1	TX_EV_TIMEOUT	Timeout controlled by Tx Event FIFO.
2	RX0_EV_TIMEOUT	Timeout controlled by Receive FIFO 0.
3	RX1_EV_TIMEOUT	Timeout controlled by Receive FIFO 1.

Bit 0 – ETOC Enable Timeout Counter
0 (NO_TIMEOUT): Timeout Counter disabled.
1 (TOS_CONTROLLED): Timeout Counter enabled.
For use of timeout function with CAN FD, see Timeout Counter.

66.6.10 MCAN Timeout Counter Value Register

Name: MCAN_TOCV
Offset: 0x2C
Reset: 0x0000FFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TOC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	TOC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – TOC[15:0] Timeout Counter (cleared on write)
The Timeout Counter is decremented in multiples of CAN bit times [1…16] depending on the configuration of MCAN_TSCC.TCP. When decremented to zero, interrupt flag MCAN_IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via MCAN_TOCC.TOS.

66.6.11 MCAN Error Counter Register

Name: MCAN_ECR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

When MCAN_CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	CEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RP	REC[6:0]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TEC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – CEL[7:0] CAN Error Logging (cleared on read)
The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.

Bit 15 – RP Receive Error Passive

Value	Description
0	The Receive Error Counter is below the error passive level of 128.
1	The Receive Error Counter has reached the error passive level of 128.

Bits 14:8 – REC[6:0] Receive Error Counter
Actual state of the Receive Error Counter, values between 0 and 127.

Bits 7:0 – TEC[7:0] Transmit Error Counter
Actual state of the Transmit Error Counter, values between 0 and 255.

66.6.12 MCAN Protocol Status Register

Name: MCAN_PSR
Offset: 0x44
Reset: 0x00000707
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		TDCV[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	1	1	1

Bit	7	6	5	4	3	2	1	0
	BO	EW	EP	ACT[1:0]		LEC[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bits 22:16 – TDCV[6:0] Transmitter Delay Compensation Value
0 to 127: Position of the secondary sample point, in CAN core clock periods, defined by the sum of the measured delay from CANTX to CANRX and MCAN_TDCR.TDCO.

Bit 14 – PXE Protocol Exception Event (cleared on read)

Value	Description
0	No protocol exception event occurred since last read access
1	Protocol exception event occurred

Bit 13 – RFDF Received a CAN FD Message (cleared on read)
This bit is set independently from acceptance filtering.

Value	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received
1	Message in CAN FD format with FDF flag set has been received

Bit 12 – RBRS BRS Flag of Last Received CAN FD Message (cleared on read)
This bit is set together with RFDF, independently from acceptance filtering.

Value	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set.

Bit 11 – RESI ESI Flag of Last Received CAN FD Message (cleared on read)
This bit is set together with RFDF, independently from acceptance filtering.

Value	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

Bits 10:8 – DLEC[2:0] Data Phase Last Error Code (set to 111 on read)
Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

Bit 7 – BO Bus_Off Status

Value	Description
0	The MCAN is not Bus_Off.
1	The MCAN is in Bus_Off state.

Bit 6 – EW Warning Status

Value	Description
0	Both error counters are below the Error_Warning limit of 96.
1	At least one of error counter has reached the Error_Warning limit of 96.

Bit 5 – EP Error Passive

Value	Description
0	The MCAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The MCAN is in the Error_Passive state.

Bits 4:3 – ACT[1:0] Activity
Monitors the CAN communication state of the CAN module.

Value	Name	Description
0	SYNCHRONIZING	Node is synchronizing on CAN communication
1	IDLE	Node is neither receiver nor transmitter
2	RECEIVER	Node is operating as receiver
3	TRANSMITTER	Node is operating as transmitter

Bits 2:0 – LEC[2:0] Last Error Code (set to 111 on read)
The LEC indicates the type of the last error to occur on the CAN bus. This field is cleared when a message has been transferred (reception or transmission) without error.

Value	Name	Description
0	NO_ERROR	No error occurred since LEC has been reset by successful reception or transmission.
1	STUFF_ERROR	More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
2	FORM_ERROR	A fixed format part of a received frame has the wrong format.
3	ACK_ERROR	The message transmitted by the MCAN was not acknowledged by another node.
4	BIT1_ERROR	During transmission of a message (with the exception of the arbitration field), the device tried to send a recessive level (bit of logical value ‘1’), but the monitored bus value was dominant.
5	BIT0_ERROR	During transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device tried to send a dominant level (data or identifier bit logical value ‘0’), but the monitored bus value was recessive. During Bus_Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the processor to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
6	CRC_ERROR	The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match the CRC calculated from the received data.
7	NO_CHANGE	Any read access to the Protocol Status Register re-initializes the LEC to ‘7’. When the LEC shows value ‘7’, no CAN bus event was detected since the last processor read access to the Protocol Status Register.

66.6.13 MCAN Transmitter Delay Compensation Register

Name: MCAN_TDCR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TDCO[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		TDCF[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 14:8 – TDCO[6:0] Transmitter Delay Compensation Offset
0 to 127: Offset value, in CAN core clock periods, defining the distance between the measured delay from CANTX to CANRX and the secondary sample point.

Bits 6:0 – TDCF[6:0] Transmitter Delay Compensation Filter
0 to 127: defines the minimum value for the SSP position, in CAN core clock periods. Dominant edges on CANRX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO.

66.6.14 MCAN Interrupt Register

Name: MCAN_IR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

Bit	31	30	29	28	27	26	25	24
			ARA	PED	PEA	WDI	BO	EW
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EP	ELO			DRX	TOO	MRAF	TSW
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARA Access to Reserved Address

Value	Description
0	No access to reserved address occurred
1	Access to reserved address occurred

Bit 28 – PED Protocol Error in Data Phase

Value	Description
0	No protocol error in data phase
1	Protocol error in data phase detected (MCAN_PSR.DLEC differs from 0 or 7)

Bit 27 – PEA Protocol Error in Arbitration Phase

Value	Description
0	No protocol error in arbitration phase
1	Protocol error in arbitration phase detected (MCAN_PSR.LEC differs from 0 or 7)

Bit 26 – WDI Watchdog Interrupt

Value	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to missing READY.

Bit 25 – BO Bus_Off Status

Value	Description
0	Bus_Off status unchanged.
1	Bus_Off status changed.

Bit 24 – EW Warning Status

Value	Description
0	Error_Warning status unchanged.
1	Error_Warning status changed.

Bit 23 – EP Error Passive

Value	Description
0	Error_Passive status unchanged.
1	Error_Passive status changed.

Bit 22 – ELO Error Logging Overflow

Value	Description
0	CAN Error Logging Counter did not overflow.
1	Overflow of CAN Error Logging Counter occurred.

Bit 19 – DRX Message stored to Dedicated Receive Buffer
The flag is set whenever a received message has been stored into a dedicated Receive Buffer.

Value	Description
0	No Receive Buffer updated.
1	At least one received message stored into a Receive Buffer.

Bit 18 – TOO Timeout Occurred

Value	Description
0	No timeout.
1	Timeout reached.

Bit 17 – MRAF Message RAM Access Failure
The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Receive Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.
The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation mode (see Restricted Operation Mode). To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

Value	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

Bit 16 – TSW Timestamp Wraparound

Value	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

Bit 15 – TEFL Tx Event FIFO Element Lost

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 14 – TEFF Tx Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bit 13 – TEFW Tx Event FIFO Watermark Reached

Value	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

Bit 12 – TEFN Tx Event FIFO New Entry

Value	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

Bit 11 – TFE Tx FIFO Empty

Value	Description
0	Tx FIFO non-empty.
1	Tx FIFO empty.

Bit 10 – TCF Transmission Cancellation Finished

Value	Description
0	No transmission cancellation finished.
1	Transmission cancellation finished.

Bit 9 – TC Transmission Completed

Value	Description
0	No transmission completed.
1	Transmission completed.

Bit 8 – HPM High Priority Message

Value	Description
0	No high priority message received.
1	High priority message received.

Bit 7 – RF1L Receive FIFO 1 Message Lost

Value	Description
0	No Receive FIFO 1 message lost.
1	Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

Bit 6 – RF1F Receive FIFO 1 Full

Value	Description
0	Receive FIFO 1 not full.
1	Receive FIFO 1 full.

Bit 5 – RF1W Receive FIFO 1 Watermark Reached

Value	Description
0	Receive FIFO 1 fill level below watermark.
1	Receive FIFO 1 fill level reached watermark.

Bit 4 – RF1N Receive FIFO 1 New Message

Value	Description
0	No new message written to Receive FIFO 1.
1	New message written to Receive FIFO 1.

Bit 3 – RF0L Receive FIFO 0 Message Lost

Value	Description
0	No Receive FIFO 0 message lost.
1	Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero.

Bit 2 – RF0F Receive FIFO 0 Full

Value	Description
0	Receive FIFO 0 not full.
1	Receive FIFO 0 full.

Bit 1 – RF0W Receive FIFO 0 Watermark Reached

Value	Description
0	Receive FIFO 0 fill level below watermark.
1	Receive FIFO 0 fill level reached watermark.

Bit 0 – RF0N Receive FIFO 0 New Message

Value	Description
0	No new message written to Receive FIFO 0.
1	New message written to Receive FIFO 0.

66.6.15 MCAN Interrupt Enable Register

Name: MCAN_IE
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Disables the corresponding interrupt.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
			ARAE	PEDE	PEAE	WDIE	BOE	EWE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPE	ELOE			DRXE	TOOE	MRAFE	TSWE
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAE Access to Reserved Address Enable

Bit 28 – PEDE Protocol Error in Data Phase Enable

Bit 27 – PEAE Protocol Error in Arbitration Phase Enable

Bit 26 – WDIE Watchdog Interrupt Enable

Bit 25 – BOE Bus_Off Status Interrupt Enable

Bit 24 – EWE Warning Status Interrupt Enable

Bit 23 – EPE Error Passive Interrupt Enable

Bit 22 – ELOE Error Logging Overflow Interrupt Enable

Bit 19 – DRXE Message stored to Dedicated Receive Buffer Interrupt Enable

Bit 18 – TOOE Timeout Occurred Interrupt Enable

Bit 17 – MRAFE Message RAM Access Failure Interrupt Enable

Bit 16 – TSWE Timestamp Wraparound Interrupt Enable

Bit 15 – TEFLE Tx Event FIFO Event Lost Interrupt Enable

Bit 14 – TEFFE Tx Event FIFO Full Interrupt Enable

Bit 13 – TEFWE Tx Event FIFO Watermark Reached Interrupt Enable

Bit 12 – TEFNE Tx Event FIFO New Entry Interrupt Enable

Bit 11 – TFEE Tx FIFO Empty Interrupt Enable

Bit 10 – TCFE Transmission Cancellation Finished Interrupt Enable

Bit 9 – TCE Transmission Completed Interrupt Enable

Bit 8 – HPME High Priority Message Interrupt Enable

Bit 7 – RF1LE Receive FIFO 1 Message Lost Interrupt Enable

Bit 6 – RF1FE Receive FIFO 1 Full Interrupt Enable

Bit 5 – RF1WE Receive FIFO 1 Watermark Reached Interrupt Enable

Bit 4 – RF1NE Receive FIFO 1 New Message Interrupt Enable

Bit 3 – RF0LE Receive FIFO 0 Message Lost Interrupt Enable

Bit 2 – RF0FE Receive FIFO 0 Full Interrupt Enable

Bit 1 – RF0WE Receive FIFO 0 Watermark Reached Interrupt Enable

Bit 0 – RF0NE Receive FIFO 0 New Message Interrupt Enable

66.6.16 MCAN Interrupt Line Select Register

Name: MCAN_ILS
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAL Access to Reserved Address Line

Bit 28 – PEDL Protocol Error in Data Phase Line

Bit 27 – PEAL Protocol Error in Arbitration Phase Line

Bit 26 – WDIL Watchdog Interrupt Line

Bit 25 – BOL Bus_Off Status Interrupt Line

Bit 24 – EWL Warning Status Interrupt Line

Bit 23 – EPL Error Passive Interrupt Line

Bit 22 – ELOL Error Logging Overflow Interrupt Line

Bit 19 – DRXL Message stored to Dedicated Receive Buffer Interrupt Line

Bit 18 – TOOL Timeout Occurred Interrupt Line

Bit 17 – MRAFL Message RAM Access Failure Interrupt Line

Bit 16 – TSWL Timestamp Wraparound Interrupt Line

Bit 15 – TEFLL Tx Event FIFO Event Lost Interrupt Line

Bit 14 – TEFFL Tx Event FIFO Full Interrupt Line

Bit 13 – TEFWL Tx Event FIFO Watermark Reached Interrupt Line

Bit 12 – TEFNL Tx Event FIFO New Entry Interrupt Line

Bit 11 – TFEL Tx FIFO Empty Interrupt Line

Bit 10 – TCFL Transmission Cancellation Finished Interrupt Line

Bit 9 – TCL Transmission Completed Interrupt Line

Bit 8 – HPML High Priority Message Interrupt Line

Bit 7 – RF1LL Receive FIFO 1 Message Lost Interrupt Line

Bit 6 – RF1FL Receive FIFO 1 Full Interrupt Line

Bit 5 – RF1WL Receive FIFO 1 Watermark Reached Interrupt Line

Bit 4 – RF1NL Receive FIFO 1 New Message Interrupt Line

Bit 3 – RF0LL Receive FIFO 0 Message Lost Interrupt Line

Bit 2 – RF0FL Receive FIFO 0 Full Interrupt Line

Bit 1 – RF0WL Receive FIFO 0 Watermark Reached Interrupt Line

Bit 0 – RF0NL Receive FIFO 0 New Message Interrupt Line

66.6.17 MCAN Interrupt Line Enable

Name: MCAN_ILE
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

Each of the two interrupt lines to the processor can be enabled/disabled separately by programming bits EINT0 and EINT1.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							EINT1	EINT0
Access							R/W	R/W
Reset							0	0

Bit 1 – EINT1 Enable Interrupt Line 1

Value	Description
0	Interrupt line MCAN_INT1 disabled.
1	Interrupt line MCAN_INT1 enabled.

Bit 0 – EINT0 Enable Interrupt Line 0

Value	Description
0	Interrupt line MCAN_INT0 disabled.
1	Interrupt line MCAN_INT0 enabled.

66.6.18 MCAN Global Filter Configuration

Name: MCAN_GFC
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages as illustrated in Standard Message ID Filter Path and Extended Message ID Filter Path.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:4 – ANFS[1:0] Accept Non-matching Frames Standard
Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

Bits 3:2 – ANFE[1:0] Accept Non-matching Frames Extended
Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RX_FIFO_0	Accept in Rx FIFO 0
1	RX_FIFO_1	Accept in Rx FIFO 1
2-3	REJECTED	Message rejected

Bit 1 – RRFS Reject Remote Frames Standard
0 (FILTER): Filter remote frames with 11-bit standard IDs.
1 (REJECT): Reject all remote frames with 11-bit standard IDs.

Bit 0 – RRFE Reject Remote Frames Extended
0 (FILTER): Filter remote frames with 29-bit extended IDs.
1 (REJECT): Reject all remote frames with 29-bit extended IDs.

66.6.19 MCAN Standard ID Filter Configuration

Name: MCAN_SIDFC
Offset: 0x84
Reset: 0x00000000
Property: Read/Write

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages as illustrated in Standard Message ID Filter Path.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LSS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLSSA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLSSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 23:16 – LSS[7:0] List Size Standard
>128: Values greater than 128 are interpreted as 128.

Value	Description
0	No standard Message ID filter.
1-128	Number of standard Message ID filter elements.

Bits 15:2 – FLSSA[13:0] Filter List Standard Start Address
Start address of standard Message ID filter list (32-bit word address, see Message RAM Configuration).
Write FLSSA with the bits [15:2] of the 32-bit address.

66.6.20 MCAN Extended ID Filter Configuration

Name: MCAN_XIDFC
Offset: 0x88
Reset: 0x00000000
Property: Read/Write

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages as described in Extended Message ID Filter Path.

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		LSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLESA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLESA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 22:16 – LSE[6:0] List Size Extended

Value	Description
0	No extended Message ID filter.
1-64	Number of extended Message ID filter elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:2 – FLESA[13:0] Filter List Extended Start Address
Start address of extended Message ID filter list (32-bit word address, see Message RAM Configuration).
Write FLESA with the bits [15:2] of the 32-bit address.

66.6.21 MCAN Extended ID AND Mask

Name: MCAN_XIDAM
Offset: 0x90
Reset: 0x1FFFFFFF
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
				EIDM[28:24]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	EIDM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EIDM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EIDM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 28:0 – EIDM[28:0] Extended ID Mask
For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

66.6.22 MCAN High Priority Message Status

Name: MCAN_HPMS
Offset: 0x94
Reset: 0x00000000
Property: Read-only

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FLST	FIDX[6:0]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSI[1:0]		BIDX[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FLST Filter List
Indicates the filter list of the matching filter element.

Value	Description
0	Standard filter list
1	Extended filter list

Bits 14:8 – FIDX[6:0] Filter Index
Index of matching filter element. Range is 0 to MCAN_SIDFC.LSS - 1 resp. MCAN_XIDFC.LSE - 1.

Bits 7:6 – MSI[1:0] Message Storage Indicator

Value	Name	Description
0	NO_FIFO_SEL	No FIFO selected.
1	LOST	FIFO message lost.
2	FIFO_0	Message stored in FIFO 0.
3	FIFO_1	Message stored in FIFO 1.

Bits 5:0 – BIDX[5:0] Buffer Index
Index of Receive FIFO element to which the message was stored. Only valid when MSI[1] = ‘1’.

66.6.23 MCAN New Data 1

Name: MCAN_NDAT1
Offset: 0x98
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDx New Data
The register holds the New Data flags of Receive Buffers 0 to 31. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

Value	Description
0	Receive Buffer not updated
1	Receive Buffer updated from new message

66.6.24 MCAN New Data 2

Name: MCAN_NDAT2
Offset: 0x9C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDx New Data
The register holds the New Data flags of Receive Buffers 32 to 63. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

Value	Description
0	Receive Buffer not updated.
1	Receive Buffer updated from new message.

66.6.25 MCAN Receive FIFO 0 Configuration

Name: MCAN_RXF0C
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
	F0OM	F0WM[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
		F0S[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	F0SA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	F0SA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 31 – F0OM FIFO 0 Operation Mode
FIFO 0 can be operated in Blocking or in Overwrite mode (see Rx FIFOs).

Value	Description
0	FIFO 0 Blocking mode.
1	FIFO 0 Overwrite mode.

Bits 30:24 – F0WM[6:0] Receive FIFO 0 Watermark

Value	Description
0	Watermark interrupt disabled.
1-64	Level for Receive FIFO 0 watermark interrupt (MCAN_IR.RF0W).
>64	Watermark interrupt disabled.

Bits 22:16 – F0S[6:0] Receive FIFO 0 Size
The Receive FIFO 0 elements are indexed from 0 to F0S-1.

Value	Description
0	No Receive FIFO 0
1-64	Number of Receive FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:2 – F0SA[13:0] Receive FIFO 0 Start Address
Start address of Receive FIFO 0 in Message RAM (32-bit word address, see Message RAM Configuration).
Write F0SA with the bits [15:2] of the 32-bit address.

66.6.26 MCAN Receive FIFO 0 Status

Name: MCAN_RXF0S
Offset: 0xA4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							RF0L	F0F
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			F0PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F0GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F0FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 25 – RF0L Receive FIFO 0 Message Lost
This bit is a copy of interrupt flag MCAN_IR.RF0L. When MCAN_IR.RF0L is reset, this bit is also reset.
Overwriting the oldest message when MCAN_RXF0C.F0OM = ‘1’ will not set this flag.

Value	Description
0	No Receive FIFO 0 message lost
1	Receive FIFO 0 message lost, also set after write attempt to Receive FIFO 0 of size zero

Bit 24 – F0F Receive FIFO 0 Full

Value	Description
0	Receive FIFO 0 not full.
1	Receive FIFO 0 full.

Bits 21:16 – F0PI[5:0] Receive FIFO 0 Put Index
Receive FIFO 0 write index pointer, range 0 to 63.

Bits 13:8 – F0GI[5:0] Receive FIFO 0 Get Index
Receive FIFO 0 read index pointer, range 0 to 63.

Bits 6:0 – F0FL[6:0] Receive FIFO 0 Fill Level
Number of elements stored in Receive FIFO 0, range 0 to 64.

66.6.27 MCAN Receive FIFO 0 Acknowledge

Name: MCAN_RXF0A
Offset: 0xA8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
			F0AI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – F0AI[5:0] Receive FIFO 0 Acknowledge Index
After the processor has read a message or a sequence of messages from Receive FIFO 0 it has to write the buffer index of the last element read from Receive FIFO 0 to F0AI. This will set the Receive FIFO 0 Get Index MCAN_RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level MCAN_RXF0S.F0FL.

66.6.28 MCAN Receive Buffer Configuration

Name: MCAN_RXBC
Offset: 0xAC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RBSA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RBSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 15:2 – RBSA[13:0] Receive Buffer Start Address
Configures the start address of the Receive Buffers section in the Message RAM (32-bit word address, see Message RAM Configuration). Also used to reference debug messages A,B,C.
Write RBSA with the bits [15:2] of the 32-bit address.

66.6.29 MCAN Receive FIFO 1 Configuration

Name: MCAN_RXF1C
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
	F1OM	F1WM[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		F1S[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	F1SA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	F1SA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 31 – F1OM FIFO 1 Operation Mode
FIFO 1 can be operated in Blocking or in Overwrite mode (see Rx FIFOs).

Value	Description
0	FIFO 1 Blocking mode.
1	FIFO 1 Overwrite mode.

Bits 30:24 – F1WM[6:0] Receive FIFO 1 Watermark

Value	Description
0	Watermark interrupt disabled
1-64	Level for Receive FIFO 1 watermark interrupt (MCAN_IR.RF1W).
>64	Watermark interrupt disabled.

Bits 22:16 – F1S[6:0] Receive FIFO 1 Size
The elements in Receive FIFO 1 are indexed from 0 to F1S - 1.

Value	Description
0	No Receive FIFO 1
1-64	Number of elements in Receive FIFO 1.
>64	Values greater than 64 are interpreted as 64.

Bits 15:2 – F1SA[13:0] Receive FIFO 1 Start Address
Start address of Receive FIFO 1 in Message RAM (32-bit word address, see Message RAM Configuration).
Write F1SA with the bits [15:2] of the 32-bit address.

66.6.30 MCAN Receive FIFO 1 Status

Name: MCAN_RXF1S
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DMS[1:0]						RF1L	F1F
Access	R	R					R	R
Reset	0	0					0	0

Bit	23	22	21	20	19	18	17	16
			F1PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F1GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F1FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 31:30 – DMS[1:0] Debug Message Status

Value	Name	Description
0	IDLE	Idle state, wait for reception of debug messages, DMA request is cleared.
1	MSG_A	Debug message A received.
2	MSG_AB	Debug messages A, B received.
3	MSG_ABC	Debug messages A, B, C received, DMA request is set.

Bit 25 – RF1L Receive FIFO 1 Message Lost
This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.
Overwriting the oldest message when MCAN_RXF1C.F1OM = ‘1’ will not set this flag.

Value	Description
0	No Receive FIFO 1 message lost.
1	Receive FIFO 1 message lost, also set after write attempt to Receive FIFO 1 of size zero.

Bit 24 – F1F Receive FIFO 1 Full

Value	Description
0	Receive FIFO 1 not full.
1	Receive FIFO 1 full.

Bits 21:16 – F1PI[5:0] Receive FIFO 1 Put Index
Receive FIFO 1 write index pointer, range 0 to 63.

Bits 13:8 – F1GI[5:0] Receive FIFO 1 Get Index
Receive FIFO 1 read index pointer, range 0 to 63.

Bits 6:0 – F1FL[6:0] Receive FIFO 1 Fill Level
Number of elements stored in Receive FIFO 1, range 0 to 64.

66.6.31 MCAN Receive FIFO 1 Acknowledge

Name: MCAN_RXF1A
Offset: 0xB8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			F1AI[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – F1AI[5:0] Receive FIFO 1 Acknowledge Index
After the processor has read a message or a sequence of messages from Receive FIFO 1 it has to write the buffer index of the last element read from Receive FIFO 1 to F1AI. This will set the Receive FIFO 1 Get Index MCAN_RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level MCAN_RXF1S.F1FL.

66.6.32 MCAN Receive Buffer / FIFO Element Size Configuration

Name: MCAN_RXESC
Offset: 0xBC
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Receive Buffer / Receive FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Receive Buffer or Receive FIFO, only the number of bytes as configured by MCAN_RXESC are stored to the Receive Buffer resp. Receive FIFO element. The rest of the frame's data field is ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RBDS[2:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			F1DS[2:0]				F0DS[2:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 10:8 – RBDS[2:0] Receive Buffer Data Field Size

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

Bits 6:4 – F1DS[2:0] Receive FIFO 1 Data Field Size

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

Bits 2:0 – F0DS[2:0] Receive FIFO 0 Data Field Size

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48-byte data field
7	64_BYTE	64-byte data field

66.6.33 MCAN Tx Buffer Configuration

Name: MCAN_TXBC
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

The sum of TFQS and NDTB may not exceed 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Bit	31	30	29	28	27	26	25	24
		TFQM	TFQS[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			NDTB[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TBSA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TBSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 30 – TFQM Tx FIFO/Queue Mode

Value	Description
0	Tx FIFO operation.
1	Tx Queue operation.

Bits 29:24 – TFQS[5:0] Transmit FIFO/Queue Size

Value	Description
0	No Tx FIFO/Queue.
1-32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 21:16 – NDTB[5:0] Number of Dedicated Transmit Buffers

Value	Description
0	No dedicated Tx Buffers.
1-32	Number of dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

Bits 15:2 – TBSA[13:0] Tx Buffers Start Address
Start address of Tx Buffers section in Message RAM (32-bit word address, see Message RAM Configuration).
Write TBSA with the bits [15:2] of the 32-bit address.

66.6.34 MCAN Tx FIFO/Queue Status

Name: MCAN_TXFQS
Offset: 0xC4
Reset: 0x00000000
Property: Read-only

The Tx FIFO/Queue status is related to the pending Tx requests listed in register MCAN_TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (MCAN_TXBRP not yet updated).

In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.
Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TFQF	TFQPI[4:0]				
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TFFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 21 – TFQF Tx FIFO/Queue Full

Value	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

Bits 20:16 – TFQPI[4:0] Tx FIFO/Queue Put Index
Tx FIFO/Queue write index pointer, range 0 to 31.

Bits 12:8 – TFGI[4:0] Tx FIFO Get Index
Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

Bits 5:0 – TFFL[5:0] Tx FIFO Free Level
Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (MCAN_TXBC.TFQM = ‘1’).

66.6.35 MCAN Tx Buffer Element Size Configuration

Name: MCAN_TXESC
Offset: 0xC8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size MCAN_TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TBDS[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – TBDS[2:0] Tx Buffer Data Field Size

Value	Name	Description
0	8_BYTE	8-byte data field
1	12_BYTE	12-byte data field
2	16_BYTE	16-byte data field
3	20_BYTE	20-byte data field
4	24_BYTE	24-byte data field
5	32_BYTE	32-byte data field
6	48_BYTE	48- byte data field
7	64_BYTE	64-byte data field

66.6.36 MCAN Transmit Buffer Request Pending

Name: MCAN_TXBRP
Offset: 0xCC
Reset: 0x00000000
Property: Read-only

MCAN_TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding MCAN_TXBRP bit is reset.

Bit	31	30	29	28	27	26	25	24
	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TRPx Transmission Request Pending for Buffer x
Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register MCAN_TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register MCAN_TXBCR.
TXBRP bits are set only for those Tx Buffers configured via MCAN_TXBC. After a MCAN_TXBRP bit has been set, a Tx scan (see Tx Handling) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).
A cancellation request resets the corresponding transmission request pending bit of register MCAN_TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.
After a cancellation has been requested, a finished cancellation is signalled via MCAN_TXBCF.
• after successful transmission together with the corresponding MCAN_TXBTO bit.
• when the transmission has not yet been started at the point of cancellation.
• when the transmission has been aborted due to lost arbitration.
• when an error occurred during frame transmission.
In DAR mode, all transmissions are automatically cancelled if they are not successful. The corresponding MCAN_TXBCF bit is set for all unsuccessful transmissions.

Value	Description
0	No transmission request pending
1	Transmission request pending

66.6.37 MCAN Transmit Buffer Add Request

Name: MCAN_TXBAR
Offset: 0xD0
Reset: 0x00000000
Property: Read/Write

If an add request is applied for a Transmit Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this Add Request is ignored.

Bit	31	30	29	28	27	26	25	24
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ARx Add Request for Transmit Buffer x
Each Transmit Buffer has its own Add Request bit. Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the processor to set transmission requests for multiple Transmit Buffers with one write to MCAN_TXBAR. MCAN_TXBAR bits are set only for those Transmit Buffers configured via TXBC. When no Transmit scan is running, the bits are reset immediately, else the bits remain set until the Transmit scan process has completed.

Value	Description
0	No transmission request added.
1	Transmission requested added.

66.6.38 MCAN Transmit Buffer Cancellation Request

Name: MCAN_TXBCR
Offset: 0xD4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CRx Cancellation Request for Transmit Buffer x
Each Transmit Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the processor to set cancellation requests for multiple Transmit Buffers with one write to MCAN_TXBCR. MCAN_TXBCR bits are set only for those Transmit Buffers configured via TXBC. The bits remain set until the corresponding bit of MCAN_TXBRP is reset.

Value	Description
0	No cancellation pending.
1	Cancellation pending.

66.6.39 MCAN Transmit Buffer Transmission Occurred

Name: MCAN_TXBTO
Offset: 0xD8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TOx Transmission Occurred for Buffer x
Each Transmit Buffer has its own Transmission Occurred bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit of register MCAN_TXBAR.

Value	Description
0	No transmission occurred.
1	Transmission occurred.

66.6.40 MCAN Transmit Buffer Cancellation Finished

Name: MCAN_TXBCF
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFx Cancellation Finished for Transmit Buffer x

Each Transmit Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via MCAN_TXBCR. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register MCAN_TXBAR.

Value	Description
0	No transmit buffer cancellation.
1	Transmit buffer cancellation finished.

66.6.41 MCAN Transmit Buffer Transmission Interrupt Enable

Name: MCAN_TXBTIE
Offset: 0xE0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TIEx Transmission Interrupt Enable for Buffer x
Each Transmit Buffer has its own Transmission Interrupt Enable bit.

Value	Description
0	Transmission interrupt disabled
1	Transmission interrupt enable

66.6.42 MCAN Transmit Buffer Cancellation Finished Interrupt Enable

Name: MCAN_TXBCIE
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFIEx Cancellation Finished Interrupt Enable for Transmit Buffer x
Each Transmit Buffer has its own Cancellation Finished Interrupt Enable bit.

Value	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

66.6.43 MCAN Transmit Event FIFO Configuration

Name: MCAN_TXEFC
Offset: 0xF0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

Bit	31	30	29	28	27	26	25	24
			EFWM[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			EFS[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EFSA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EFSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 29:24 – EFWM[5:0] Event FIFO Watermark

Value	Description
0	Watermark interrupt disabled.
1-32	Level for Tx Event FIFO watermark interrupt (MCAN_IR.TEFW).
>32	Watermark interrupt disabled.

Bits 21:16 – EFS[5:0] Event FIFO Size
The Tx Event FIFO elements are indexed from 0 to EFS - 1.

Value	Description
0	Tx Event FIFO disabled.
1-32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

Bits 15:2 – EFSA[13:0] Event FIFO Start Address
Start address of Tx Event FIFO in Message RAM (32-bit word address, see Message RAM Configuration).
Write EFSA with the bits [15:2] of the 32-bit address.

66.6.44 MCAN Tx Event FIFO Status

Name: MCAN_TXEFS
Offset: 0xF4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							TEFL	EFF
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				EFPI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				EFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			EFFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 25 – TEFL Tx Event FIFO Element Lost
This bit is a copy of interrupt flag MCAN_IR.TEFL. When MCAN_IR.TEFL is reset, this bit is also reset.

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 24 – EFF Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bits 20:16 – EFPI[4:0] Event FIFO Put Index
Tx Event FIFO write index pointer, range 0 to 31.

Bits 12:8 – EFGI[4:0] Event FIFO Get Index
Tx Event FIFO read index pointer, range 0 to 31.

Bits 5:0 – EFFL[5:0] Event FIFO Fill Level
Number of elements stored in Tx Event FIFO, range 0 to 32.

66.6.45 MCAN Tx Event FIFO Acknowledge

Name: MCAN_TXEFA
Offset: 0xF8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				EFAI[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – EFAI[4:0] Event FIFO Acknowledge Index
After the processor has read an element or a sequence of elements from the Tx Event FIFO, it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index MCAN_TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level MCAN_TXEFS.EFFL.

67. Timer Counter (TC)

67.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

67.2 Embedded Characteristics

- Total of six Channels
- 32-bit Channel Size
- Wide Range of Functions Including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder with real time filtering reports
  - 2-bit Gray up/down count for stepper motor
- Each Channel is User-Configurable and Contains:
  - Three external clock inputs
  - Five internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Interrupt Line
- Read of the Capture Registers by the DMAC
- Compare and Input Edge Trigger Events for DMA
- Compare Event Fault Generation for PWM
- Register Write Protection
- Safety/Security Reports

67.3 Block Diagram

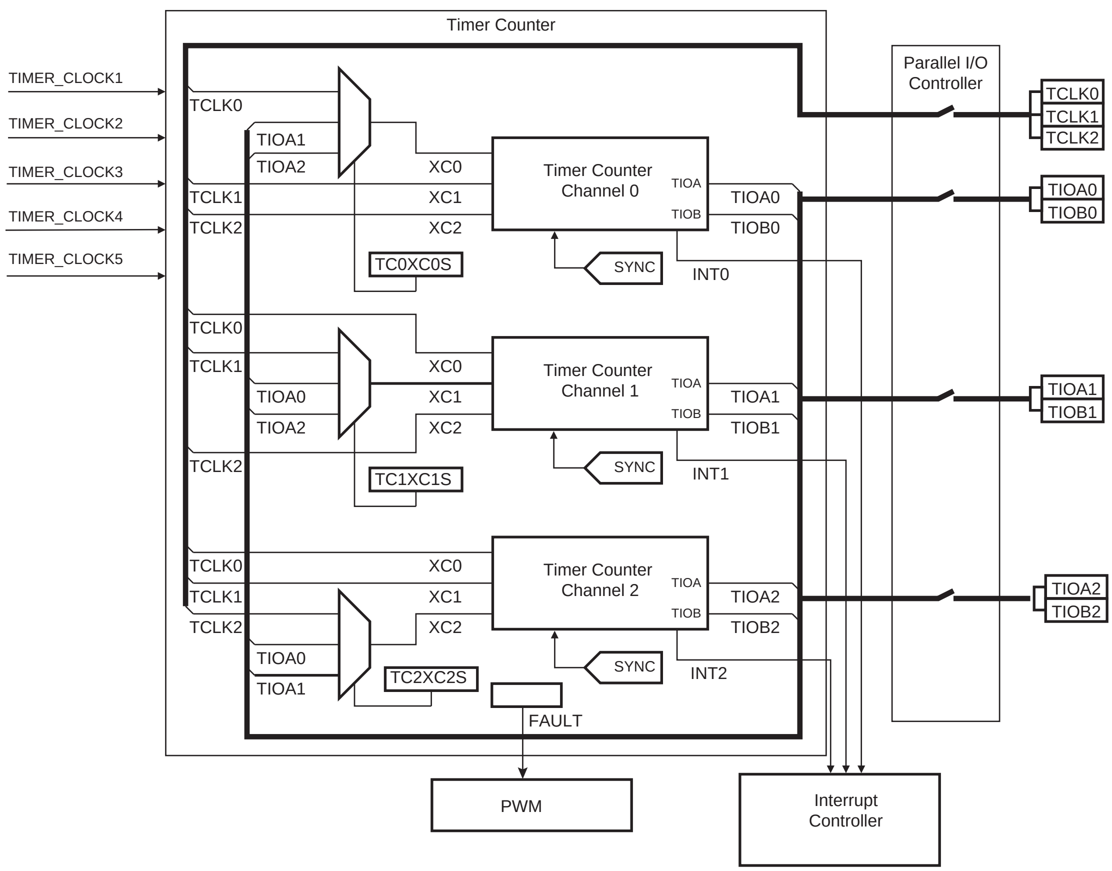
Table 67-1. Timer Counter Clock Assignment

Name	Definition
TIMER_CLOCK1	GCLK [TC_ID]
TIMER_CLOCK2	MCK/8
TIMER_CLOCK3	MCK/32
TIMER_CLOCK4	MCK/128
TIMER_CLOCK5 (See Note)	TD_SLCK

Note:

1. The GCLK[TC_ID] frequency must be at least three times lower than peripheral clock frequency.

Figure 67-1. TC Block Diagram



Note:

The QDEC connections are detailed in the figure Predefined Connection of the Quadrature Decoder with Timer Counters.

Table 67-2. Channel Signal Description

Signal Name	Description
XC0, XC1, XC2	Channel clock source that can be connected to TIOAx, TIOBx, TCLKx
TIMER_CLOCK1-5	Channel clock source from system clocks

...........continued	
Signal Name	**Description**
TIOAx	Capture mode: Timer Counter input Waveform mode: Timer Counter output
TIOBx	Capture mode: Timer Counter input Waveform mode: Timer Counter input/output
INT	Interrupt signal output (internal signal)
SYNC	Synchronization input signal (from Configuration register)

67.4 Pin List

Table 67-3. Pin List

Pin Name	Description	Type
TCLK0–TCLK2	External clock input	Input
TIOA0–TIOA2	I/O line A	I/O
TIOB0–TIOB2	I/O line B	I/O

Note: This table provides pin names of a first instance of a Timer Counter block. For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5” are the external clock input pins of a second Timer Counter block.

67.5 Product Dependencies

67.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the TC pins to their peripheral functions.

67.5.2 Power Management

The TC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the Timer Counter clock of each channel.

67.5.3 Interrupt Sources

The TC has an interrupt line per channel connected to the interrupt controller. Handling the TC interrupt requires programming the interrupt controller before configuring the TC.

67.5.4 Synchronization Inputs from PWM

The TC has trigger/capture inputs internally connected to the PWM. See Synchronization with PWM and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

67.5.5 Fault Output

The TC has the FAULT output internally connected to the fault input of PWM. See Fault Mode and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

67.6 Functional Description

67.6.1 Description

All channels of the Timer Counter are independent and identical in operation except when the QDEC is enabled. The registers for channel programming are listed in Register Summary.

67.6.2 32-bit Counter

Each 32-bit channel is organized around a 32-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 2^{32}-1 and passes to zero, an overflow occurs and the COVFS bit in the Interrupt Status register (TC_SR) is set.

The current value of the counter is accessible in real time by reading the Counter Value register (TC_CV). The counter can be reset by a trigger. In this case, the counter value passes to zero on the next valid edge of the selected clock.

67.6.3 Clock Selection

Input clock signals of each channel can be connected either to the external inputs TCLKx, or to the internal I/O signals TIOAx for chaining(1) by programming the Block Mode register (TC_BMR). See the figure Clock Chaining Selection.

Each channel can independently select a source for its counter(2):

- Signals from other channels: XC0, XC1 or XC2
- Signals from the system: GCLK [TC_ID], MCK/8, MCK/32, MCK/128, TD_SLCK

This selection is made by the TCCLKS bits in the Channel Mode register (TC_CMRx).

The selected clock can be inverted with TC_CMRx.CLKI. This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the TC_CMRx defines this signal (none, XC0, XC1, XC2). See the figure Clock Selection.

Notes:

1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   - Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMRx.ASWTRG.
   - Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous internal clock GCLK[TC_ID] is used, the duration of each of its levels must be longer than the peripheral clock period, so the clock frequency will be at least 2.5 times lower than the peripheral clock.

Figure 67-2. Clock Chaining Selection

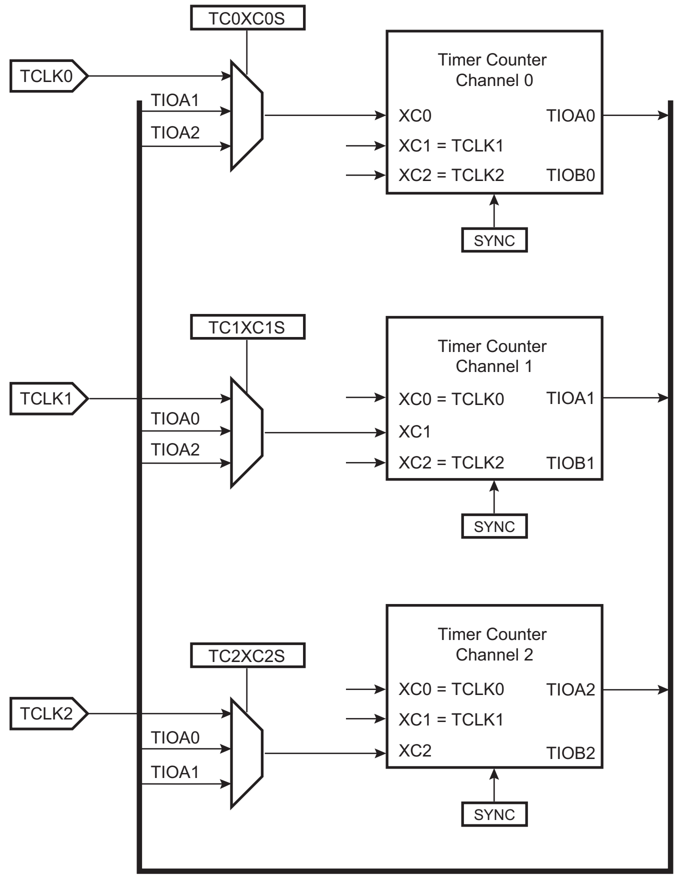
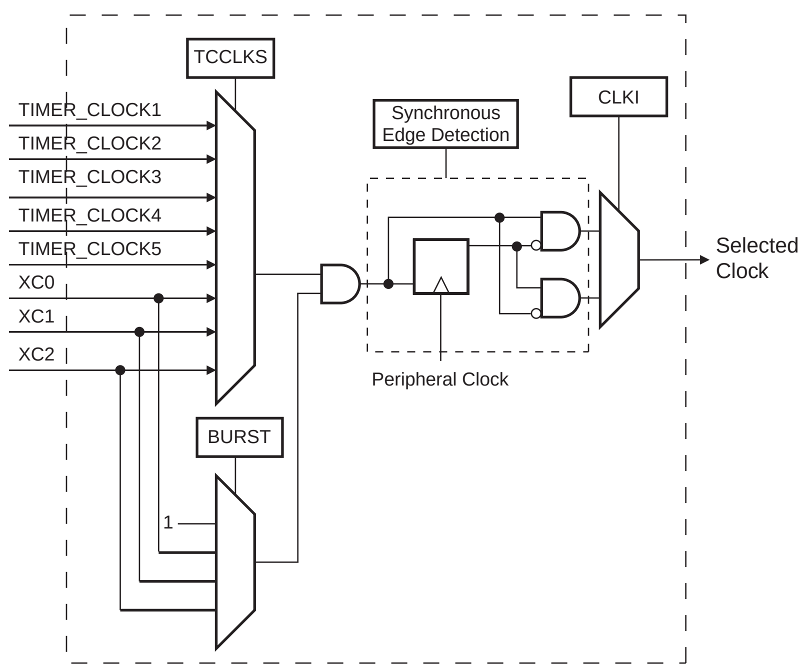


Figure 67-3. Clock Selection

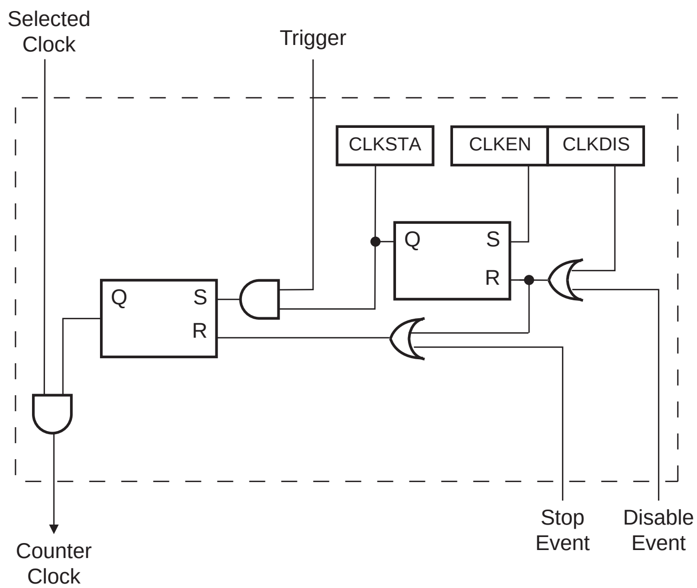


67.6.4 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped, as shown in the following figure.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the Channel Control register (TC_CCR). In Capture mode it can be disabled by an RB load event if TC_CMRx.LDBDIS is set to ‘1’. In Waveform mode, it can be disabled by an RC Compare event if TC_CMRx.CPCDIS is set to ‘1’. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can reenable the clock. When the clock is enabled, TC_SR.CLKSTA is set.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (TC_CMRx.LDBSTOP = 1) or an RC compare event in Waveform mode (TC_CMRx.CPCSTOP = 1). The start and the stop commands are effective only if the clock is enabled.

Figure 67-4. Clock Control



67.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with TC_CMRx.WAVE.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

67.6.6 Trigger Events

Input Events

An input trigger event resets the internal counter value and starts the counter clock. Three types of trigger are common to both modes, and an external trigger is available to each mode (Capture or Waveform).

Regardless of the input trigger event used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger event, especially when a low frequency signal is selected as the clock.

The following input trigger events are common to both modes:

- Software trigger: Each channel has a software trigger, available by setting TC_CCR.SWTRG.

- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR with SYNC set.
- Compare RC trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if TC_CMRx.CPCTRG is set.

The timer channel can also be configured to be triggered by an external event.

In Capture mode, the external trigger signal can be selected between TIOAx and TIOBx.

In Waveform mode, an external event can be programmed on one of the following signals: TIOBx, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting TC_CMRx.ENETRG.

If an external trigger event is used, the duration of the pulses must be longer than the peripheral clock period to be detected.

Output Events

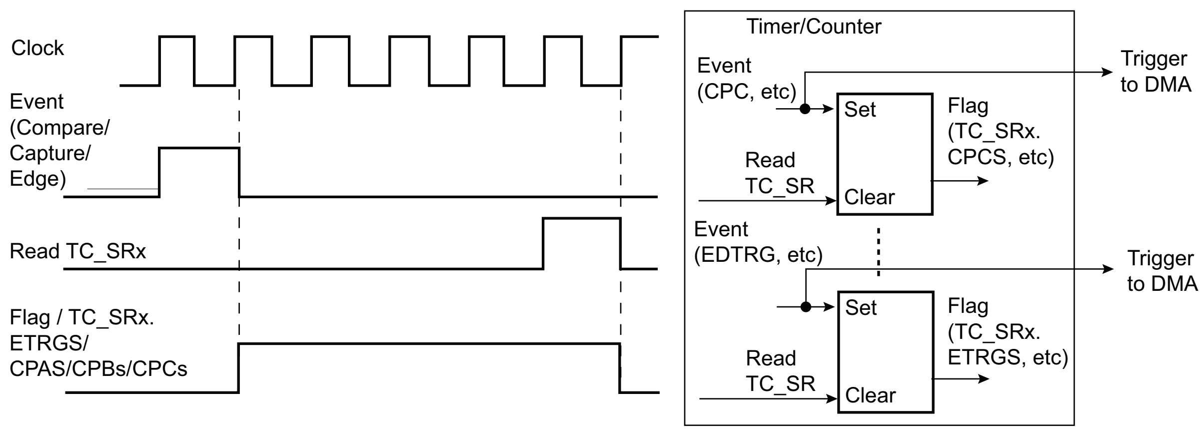
The internal counter and associated comparators provide generic trigger events for DMA transfers. These trigger events are selected in the DMA peripheral (refer to the section “DMA Controller”).

Three events correspond to the capture and compare events for which status flags CPAS, CPBS and CPCS are provided in TC_SRx.

One event corresponds to the detection of an edge on either the TIOA, TIOB or TCLK0/1/2 input depending on operating mode and configuration, for which the ETRGS status flag is provided in TC_SRx.

The status flags are cleared by software whereas the compare events setting the corresponding flags are pulses of one clock period only.

Figure 67-5. TC Output Trigger Events for DMA



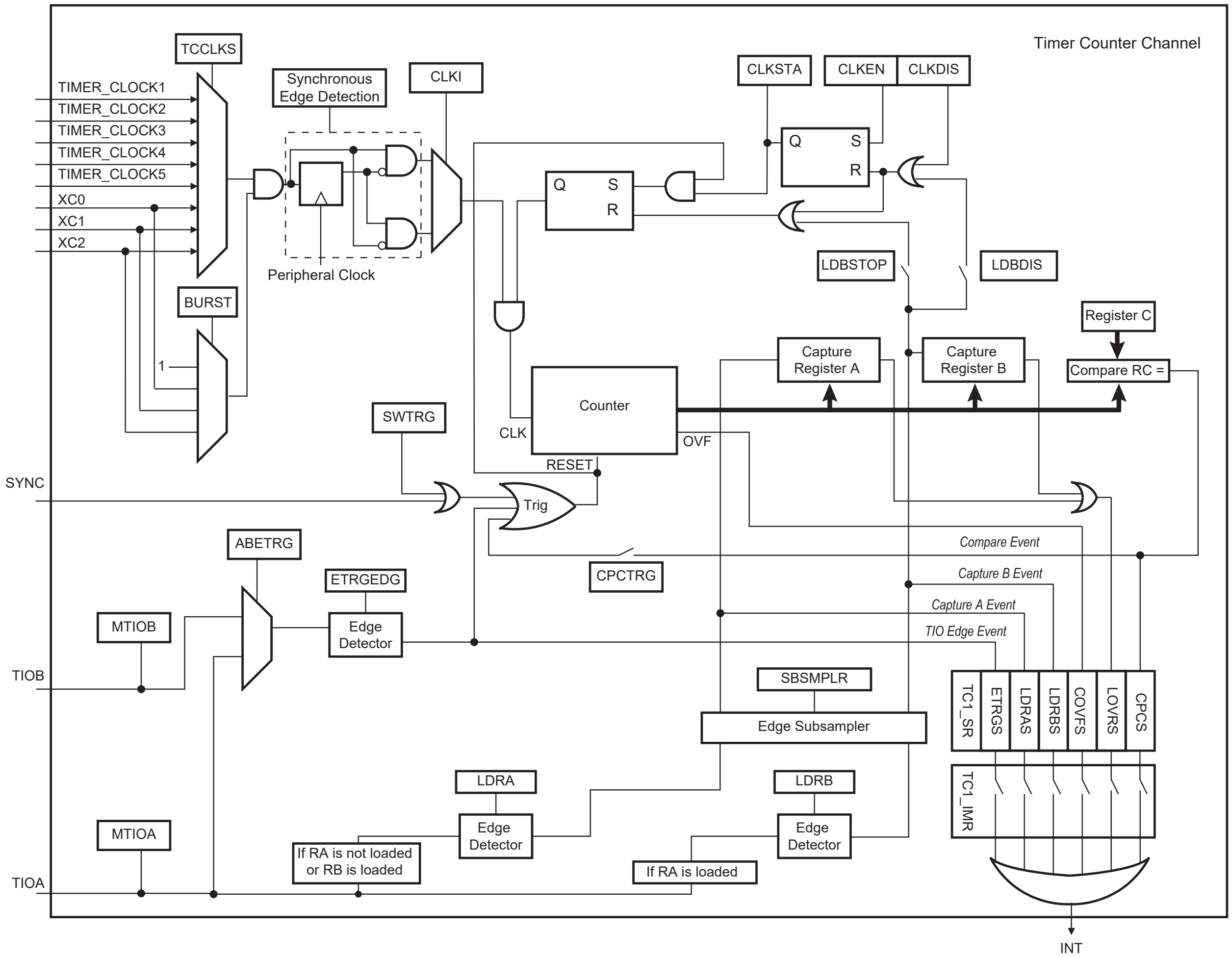
When triggered by the TC output events, the DMA can perform any access allowed in the system memory, including accesses to TC when configured in Capture mode (see Transferring Timer Values with DMAC in Capture Mode).

67.6.7 Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

The ABETRG bit in the TC_CMR selects TIOAx or TIOBx input signal as an external trigger or the trigger signal from the output comparator of the PWM module. The External Trigger Edge Selection parameter (ETRGEDG field in TC_CMR) defines the edge (rising, falling, or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

Figure 67-6. Capture Mode



67.6.8 Capture Mode

Capture mode is entered by clearing TC_CMRx.WAVE.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOAx and TIOBx signals which are considered as inputs.

The figure Capture Mode shows the configuration of the TC channel when programmed in Capture mode.

67.6.9 Capture Registers A and B

Registers A and B (TC_RA and TC_RB) are used as capture registers. They can be loaded with the counter value when a programmable event occurs on the signal TIOAx.

TC_CMRx.LDRA defines the TIOAx selected edge for the loading of TC_RA, and TC_CMRx.LDRB defines the TIOAx selected edge for the loading of TC_RB.

The subsampling ratio defined by TC_CMRx.SBSMPLR is applied to these selected edges, so that the loading of Register A and Register B occurs once every 1, 2, 4, 8 or 16 selected edges.

TC_RA is loaded only if it has not been loaded since the last trigger or if TC_RB has been loaded since the last loading of TC_RA.

TC_RB is loaded only if TC_RA has been loaded since the last trigger or the last loading of TC_RB.

Loading TC_RA or TC_RB before the read of the last value loaded sets TC_SR.LOVRS. In this case, the old value is overwritten.

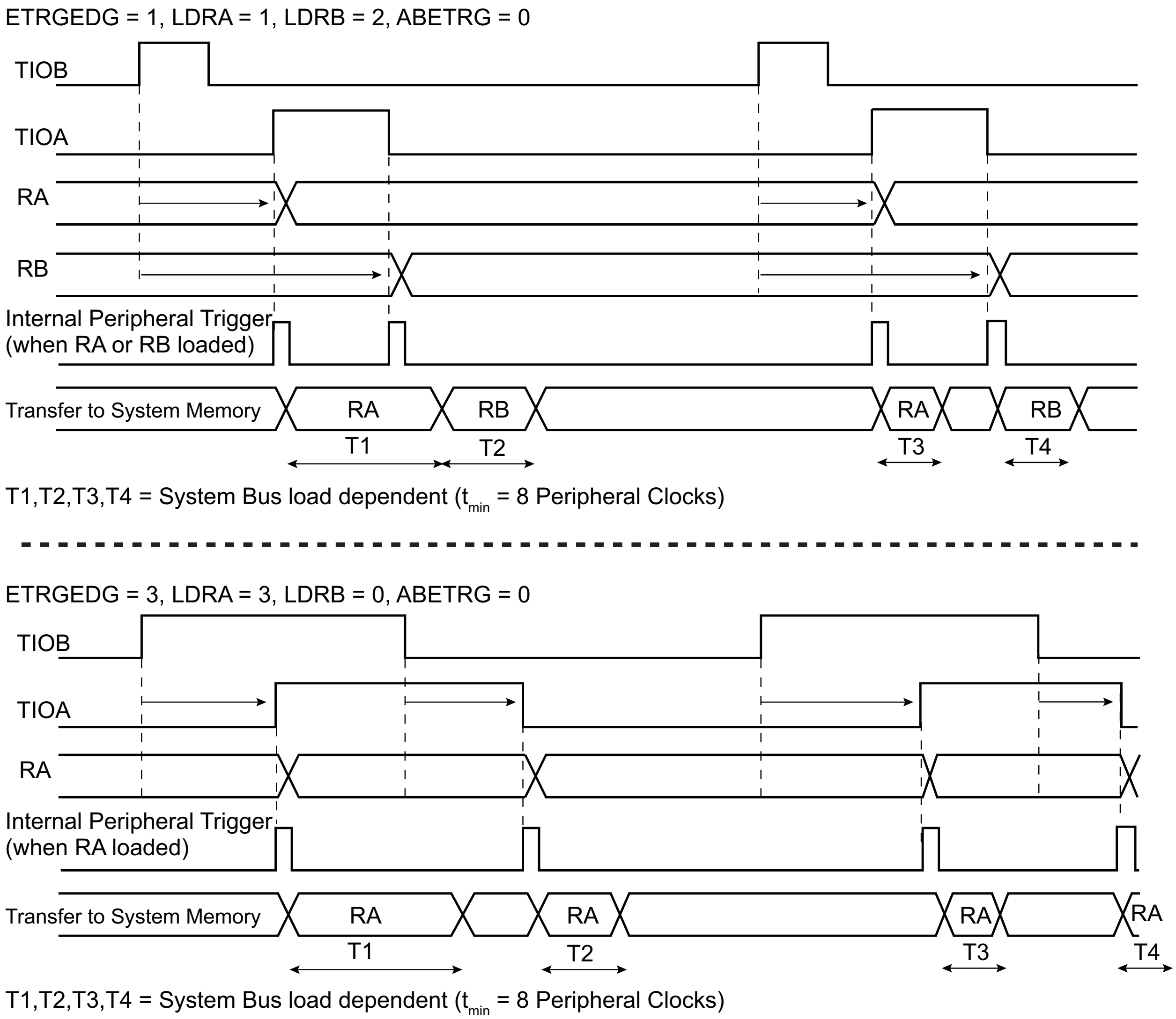
When DMA is used, the Register AB (TC_RAB) address must be configured as source address of the transfer. TC_RAB provides the next unread value from TC_RA and TC_RB. It may be read by the DMA after a request has been triggered upon loading TC_RA or TC_RB.

67.6.10 Transferring Timer Values with DMAC in Capture Mode

The DMAC can perform access from the TC to system memory in Capture mode only.

The following figure illustrates how TC_RA and TC_RB can be loaded in the system memory without processor intervention.

Figure 67-7. Example of Transfer with DMAC in Capture Mode



67.6.11 Waveform Mode

Waveform mode is entered by setting the TC_CMRx.WAVE bit.

In Waveform mode, the TC channel generates one or two PWM signals with the same frequency and independently programmable duty cycles, or generates different types of one-shot or repetitive pulses.

In this mode, TIOAx is configured as an output and TIOBx is defined as an output if it is not used as an external event (EEVT parameter in TC_CMR).

The figure Waveform Mode shows the configuration of the TC channel when programmed in Waveform operating mode.

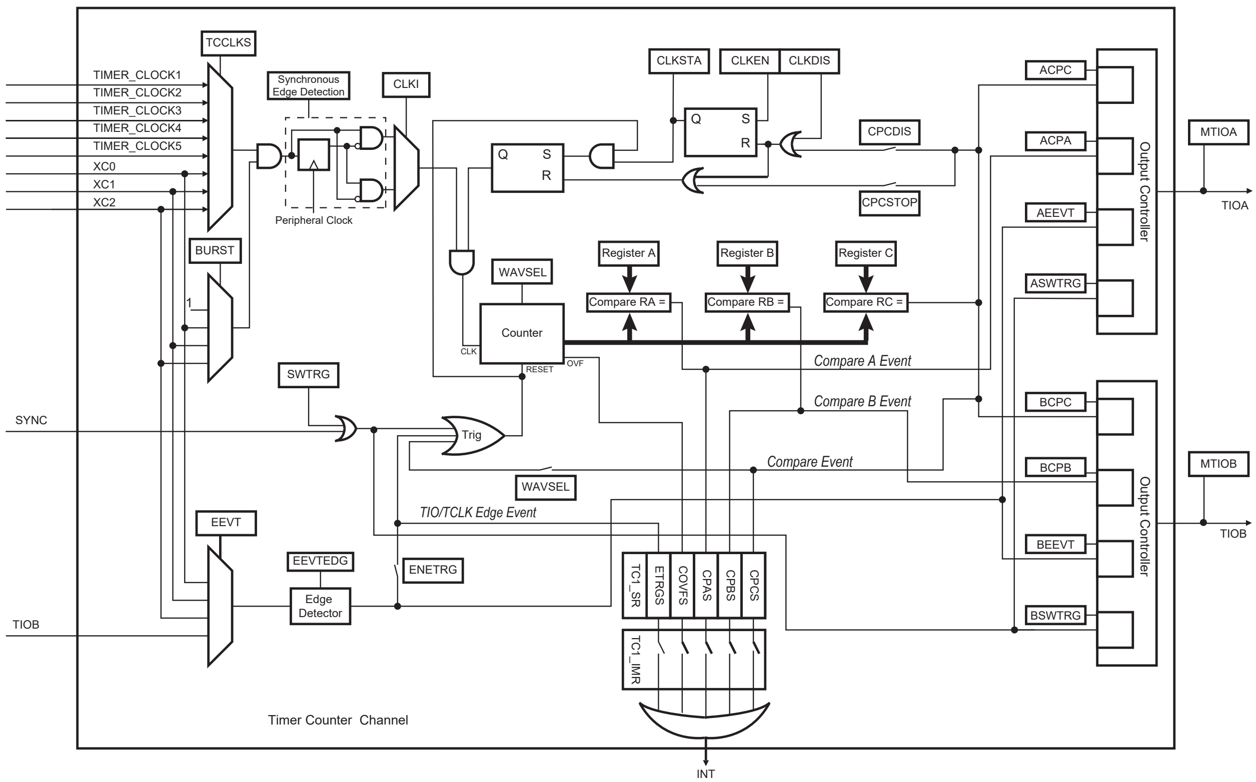
67.6.12 Waveform Selection

Depending on the WAVSEL parameter in TC_CMR, the behavior of TC_CV varies.

With any selection, TC_RA, TC_RB and TC_RC can all be used as compare registers.

RA Compare is used to control the TIOAx output, RB Compare is used to control the TIOBx output (if correctly configured) and RC Compare is used to control TIOAx and/or TIOBx outputs.

Figure 67-8. Waveform Mode



67.6.12.1 WAVSEL = 00

When WAVSEL = 00, the value of TC_CV is incremented from 0 to 2^{32}-1. Once 2^{32}-1 has been reached, the value of TC_CV is reset. Incrementation of TC_CV starts again and the cycle continues.

An external event trigger or a software trigger can reset the value of TC_CV. It is important to note that the trigger may occur at any time.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration. At the same time, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Figure 67-9. WAVSEL = 00 without Trigger

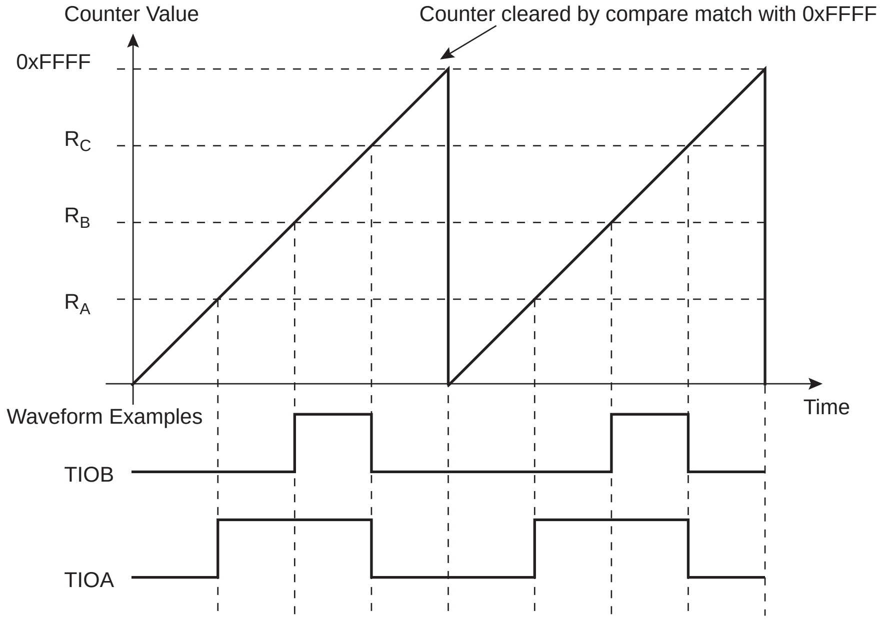
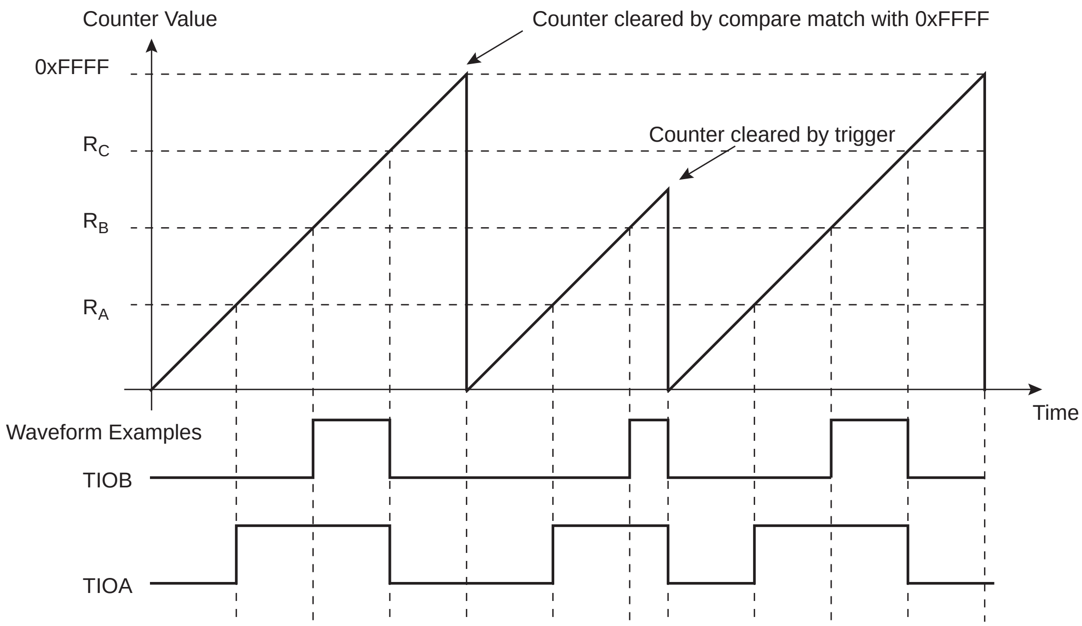


Figure 67-10. WAVSEL = 00 with Trigger



67.6.12.2 WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on.

It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly.

See the following figures.

In addition, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Figure 67-11. WAVSEL = 10 without Trigger

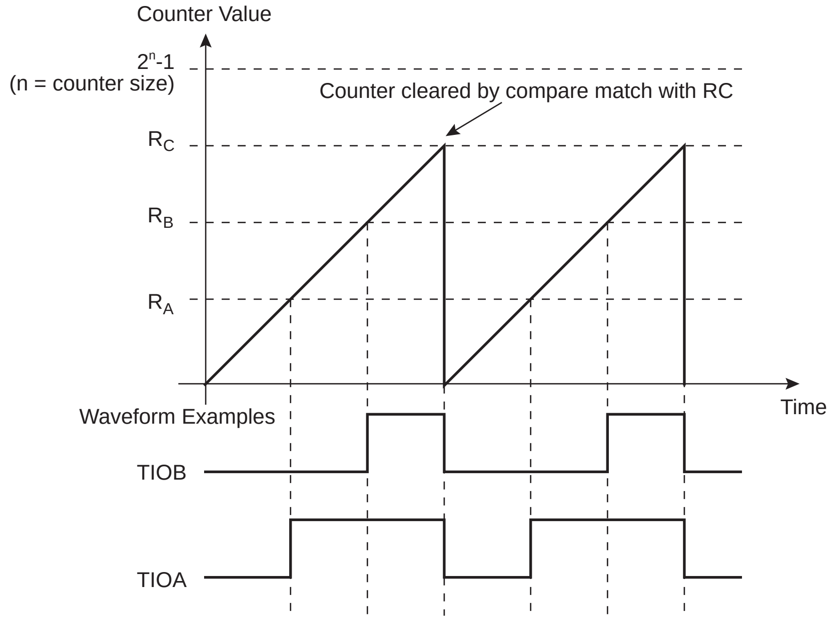
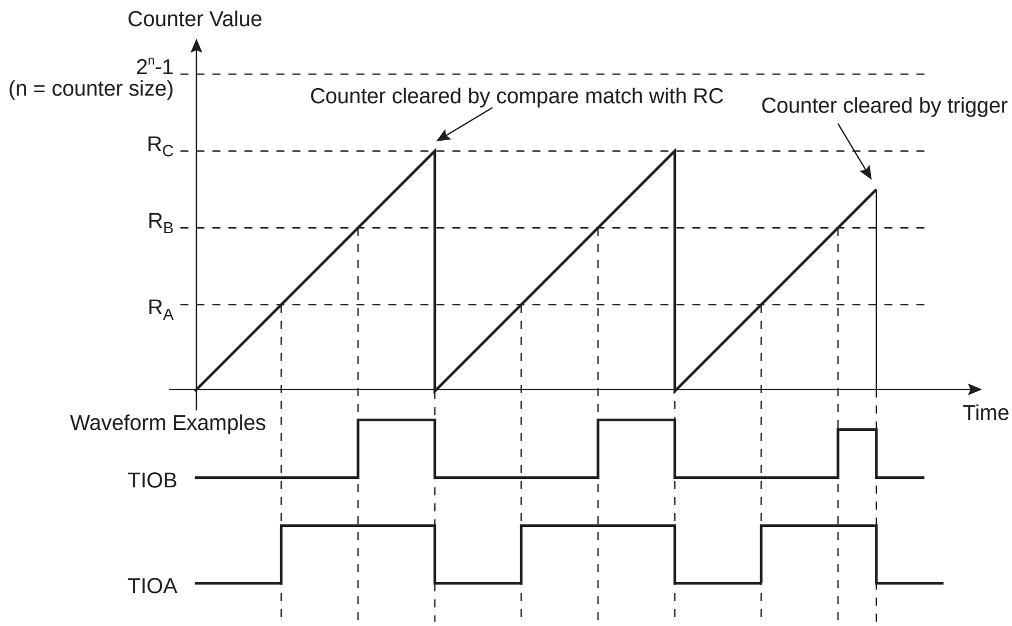


Figure 67-12. WAVSEL = 10 with Trigger



67.6.12.3 WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then reincremented to 2^{32}-1 and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 67-13. WAVSEL = 01 without Trigger

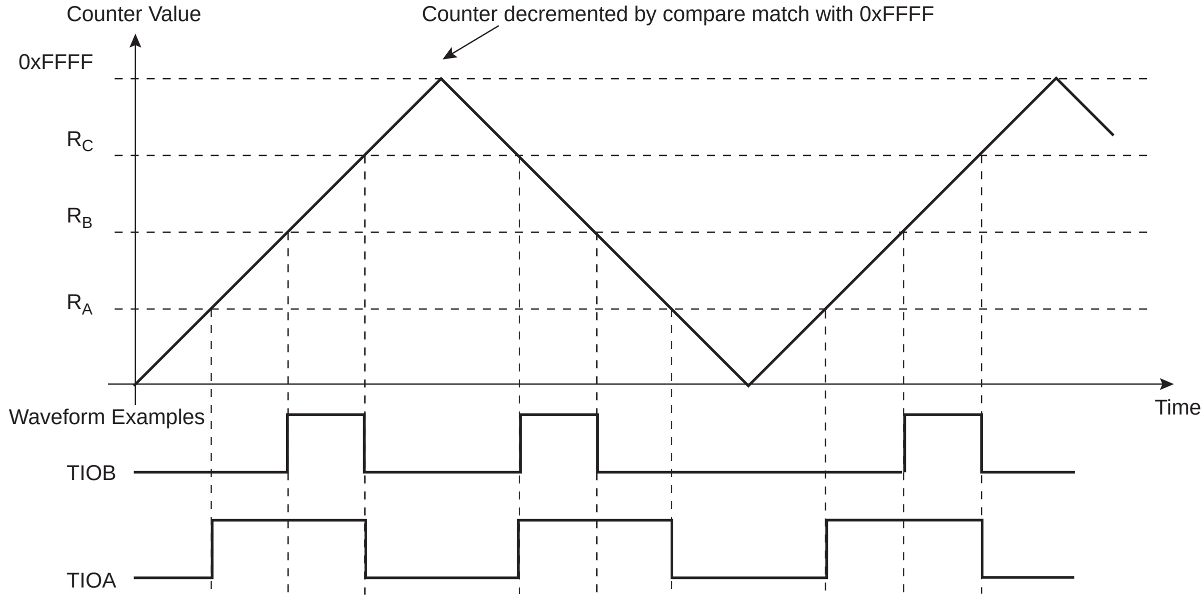
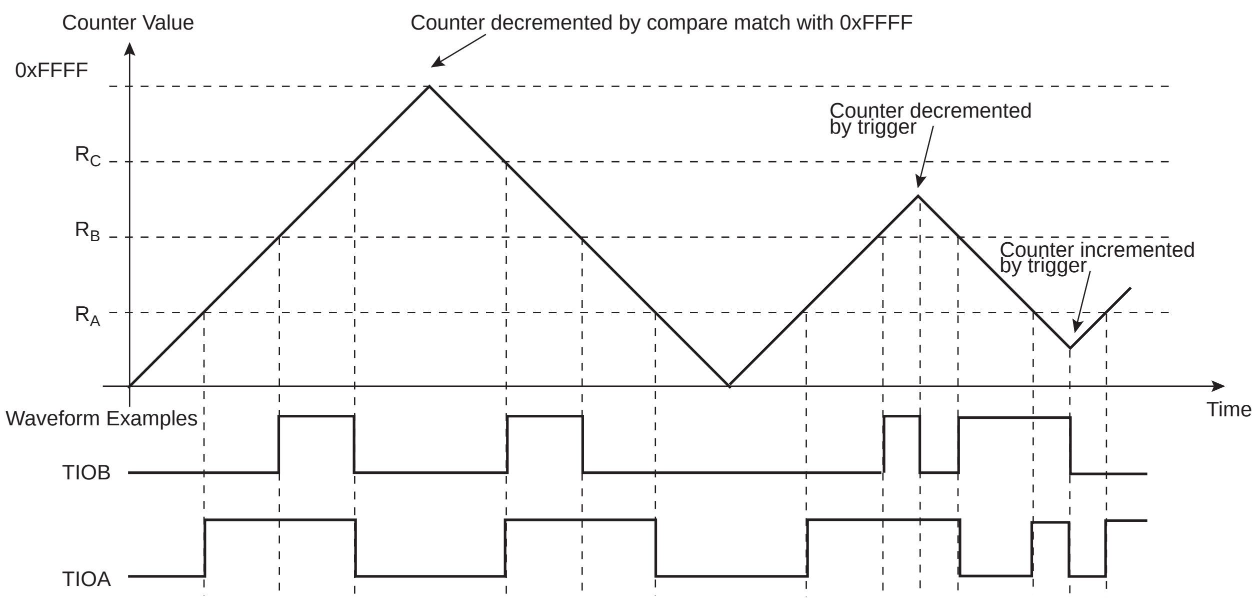


Figure 67-14. WAVSEL = 01 with Trigger



67.6.12.4 WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then reincremented to RC and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 67-15. WAVSEL = 11 without Trigger

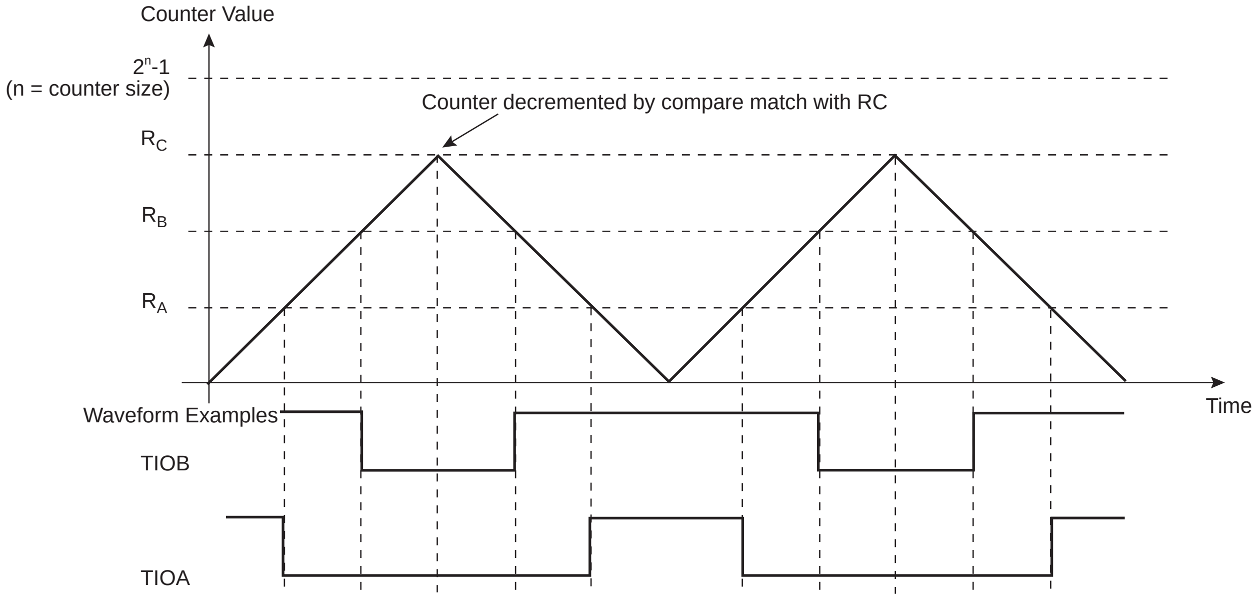
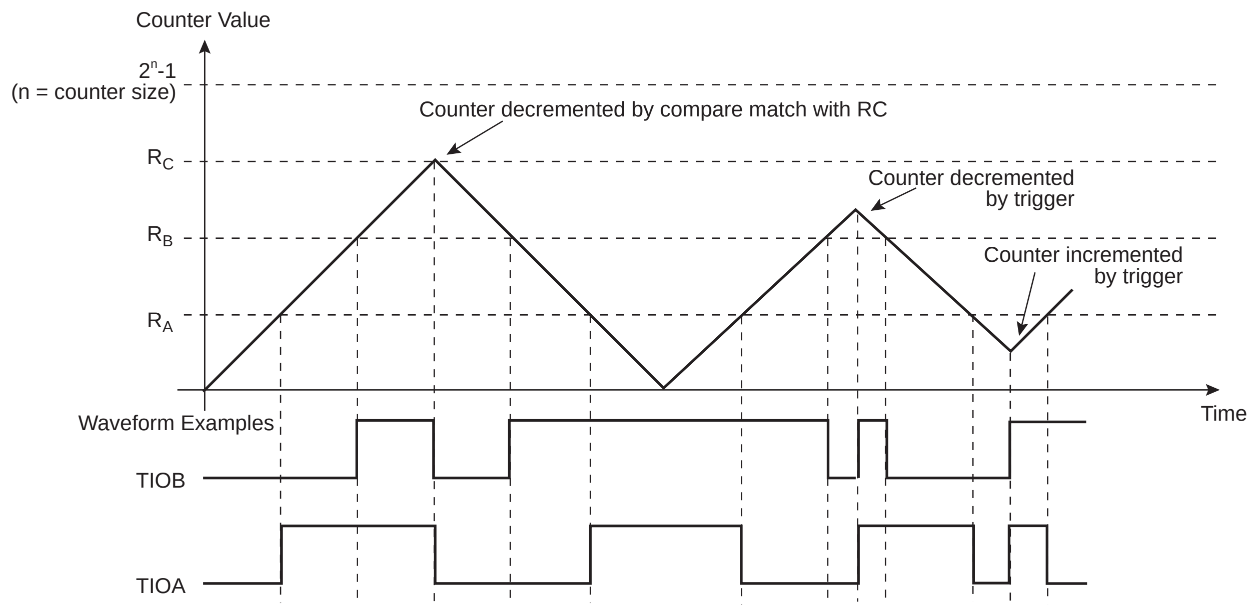


Figure 67-16. WAVSEL = 11 with Trigger



67.6.13 External Event/Trigger Conditions

An external event can be programmed to be detected on one of the clock sources (XC0, XC1, XC2) or TIOBx. The external event selected can then be used as a trigger.

The event trigger is selected using TC_CMR.EEVT. The trigger edge (rising, falling or both) for each of the possible external triggers is defined in TC_CMR.EEVTEDG. If EEVTEDG is cleared (none), no external event is defined.

If TIOBx is defined as an external event signal (EEVT = 0), TIOBx is no longer used as an output and the compare register B is not used to generate waveforms and subsequently no IRQs. In this case, the TC channel can only generate a waveform on TIOAx.

When an external event is defined, it can be used as a trigger by setting TC_CMR.ENETRG.

As in Capture mode, the SYNC signal and the software trigger are also available as triggers. RC Compare can also be used as a trigger depending on the parameter WAVSEL.

67.6.14 Synchronization with PWM

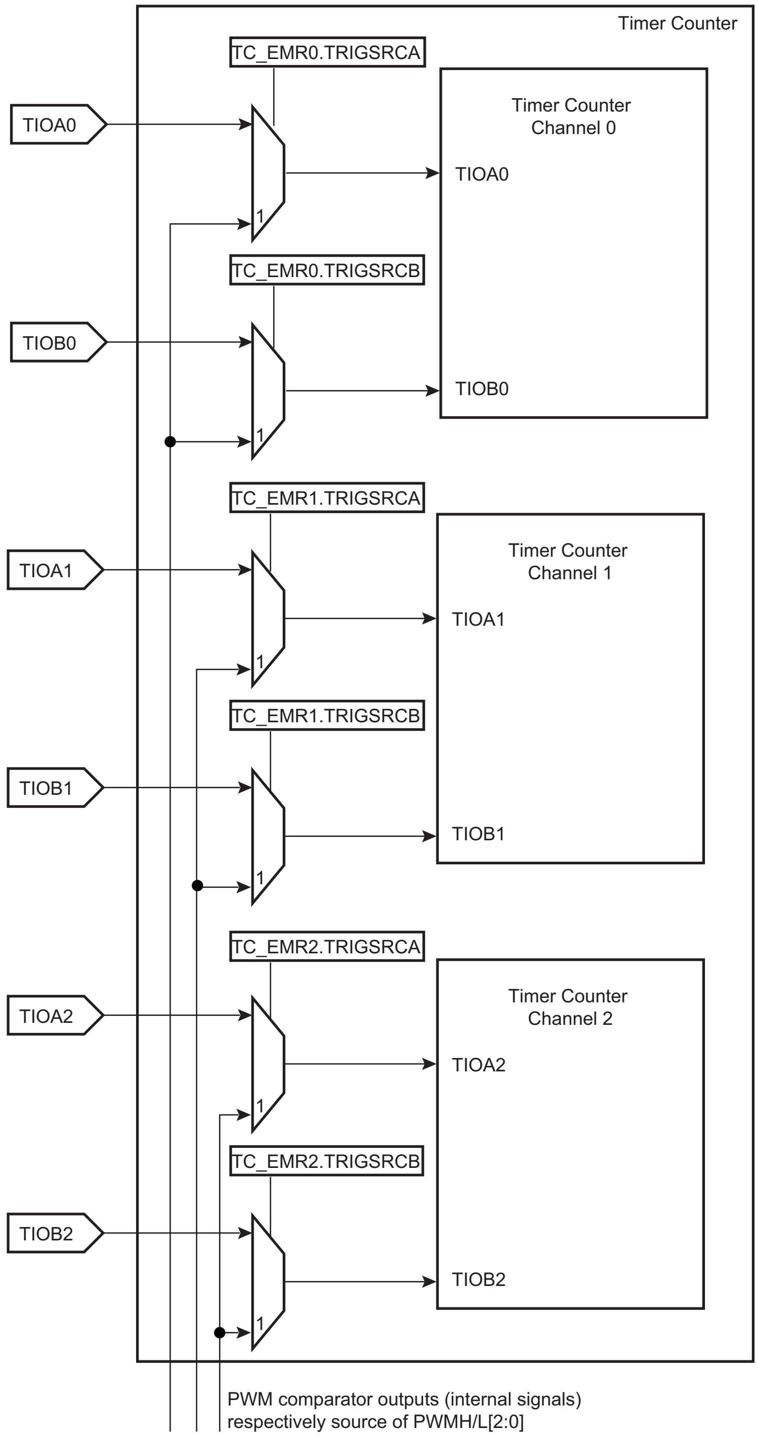
The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

PWM comparator outputs (internal signals without dead-time insertion - OCx), respectively source of the PWMH/L[2:0] outputs, are routed to the internal TC inputs. These specific TC inputs are multiplexed with TIOA/B input signal to drive the internal trigger/capture events.

The selection is made in the Extended Mode register (TC_EMR) fields TRIGSRCA and TRIGSRCB (see TC_EMRx).

Each channel of the TC module can be synchronized by a different PWM channel as described in the following figure.

Figure 67-17. Synchronization with PWM



67.6.15 Output Controller

The output controller defines the output level changes on TIOAx and TIOBx following an event. TIOBx control is used only if TIOBx is defined as output (not as an external event).

The following events control TIOAx and TIOBx:

- Software trigger
- External event

- RC compare

RA Compare controls TIOAx, and RB Compare controls TIOBx. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

67.6.16 Quadrature Decoder

67.6.16.1 Description

The quadrature decoder (QDEC) is driven by TIOA0, TIOB0 and TIOB1 input pins and drives the timer counter of channel 0 and 1. Channel 2 can be used as a time base in case of speed measurement requirements (see the following figure).

When writing a ‘0’ to TC_BMR.QDEN, the QDEC is bypassed and the IO pins are directly routed to the timer counter function.

TIOA0 and TIOB0 are to be driven by the two dedicated quadrature signals from a rotary sensor mounted on the shaft of the off-chip motor.

A third signal from the rotary sensor can be processed through pin TIOB1 and is typically dedicated to be driven by an index signal if it is provided by the sensor. This signal is not required to decode the quadrature signals PHA, PHB.

TC_CMRx.TCCLKS must be configured to select XC0 input (i.e., 0x101). Field TC0XC0S has no effect as soon as the QDEC is enabled.

Either speed or position/revolution can be measured. Position channel 0 accumulates the edges of PHA, PHB input signals giving a high accuracy on motor position whereas channel 1 accumulates the index pulses of the sensor, therefore the number of rotations. Concatenation of both values provides a high level of precision on motion system position.

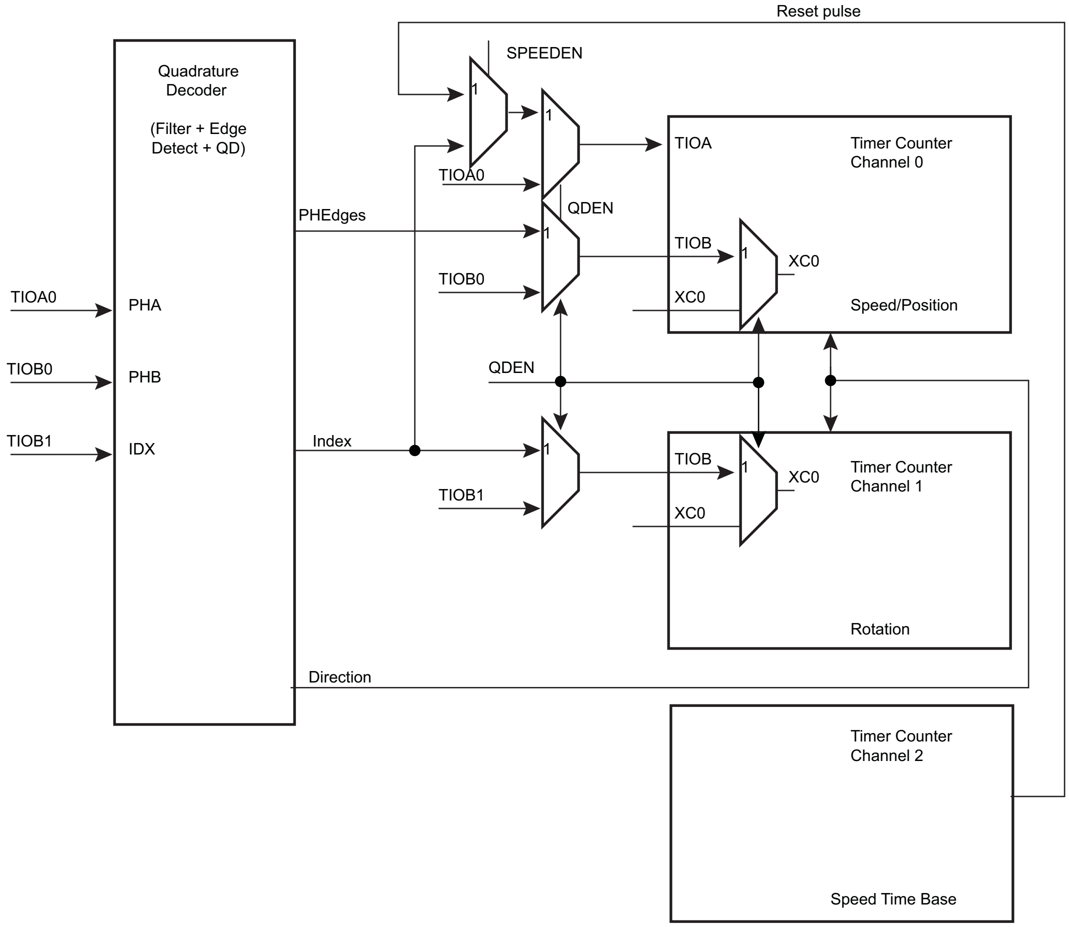
In Speed mode, position cannot be measured but revolution can be measured.

Inputs from the rotary sensor can be filtered prior to downstream processing. Accommodation of input polarity, phase definition and other factors are configurable.

Interruptions can be generated on different events.

A compare function (using TC_RC) is available on channel 0 (speed/position) or channel 1 (rotation) and can generate an interrupt by means of TC_SRx.CPCS.

Figure 67-18. Predefined Connection of the Quadrature Decoder with Timer Counters



67.6.16.2 Input Preprocessing

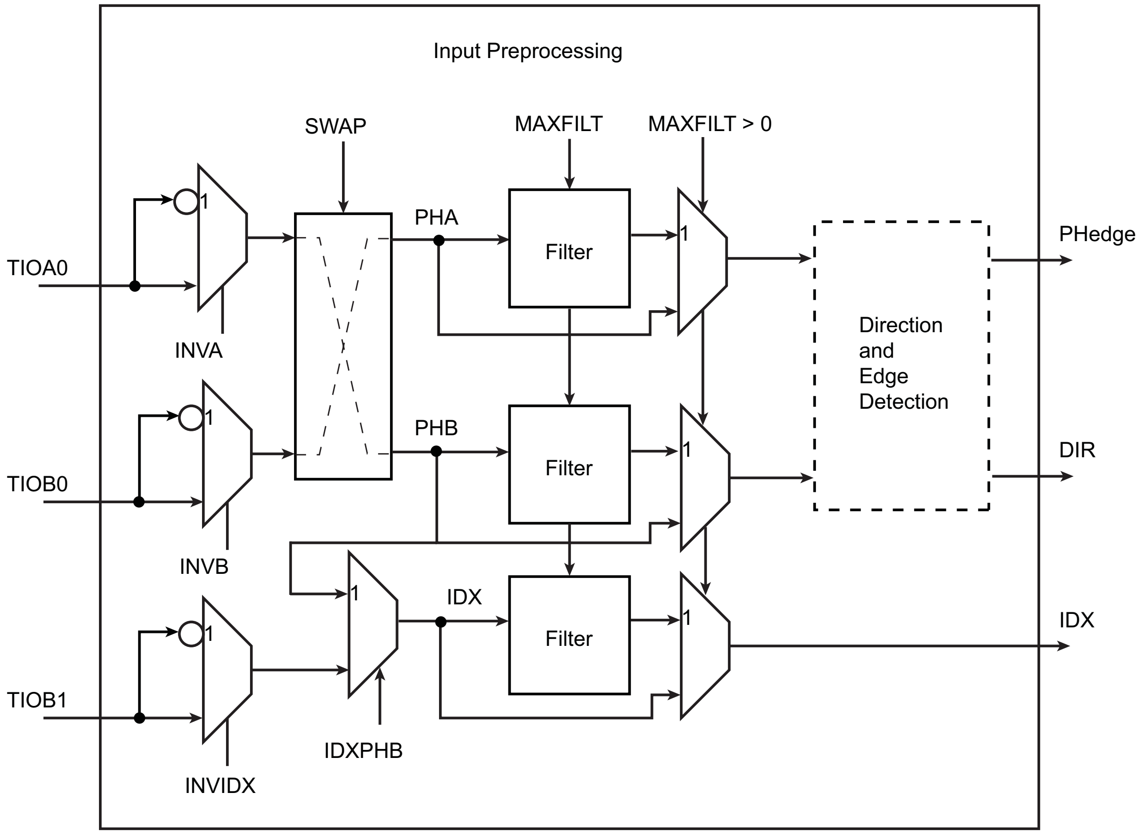
Input preprocessing consists of capabilities to take into account rotary sensor factors such as polarities and phase definition followed by configurable digital filtering.

Each input can be negated and swapping PHA, PHB is also configurable.

TC_BMR. MAXFILT is used to configure a minimum duration for which the pulse is stated as valid. When the filter is active, pulses with a duration lower than (MAXFILT +1) × $t_{peripheral\ clock}$ are not passed to downstream logic.

The value of (MAXFILT +1) × $t_{peripheral\ clock}$ must not be greater than 10% of the minimum pulse on PHA, PHB or index when the rotary encoder speed is at its maximum. This speed depends on the application.

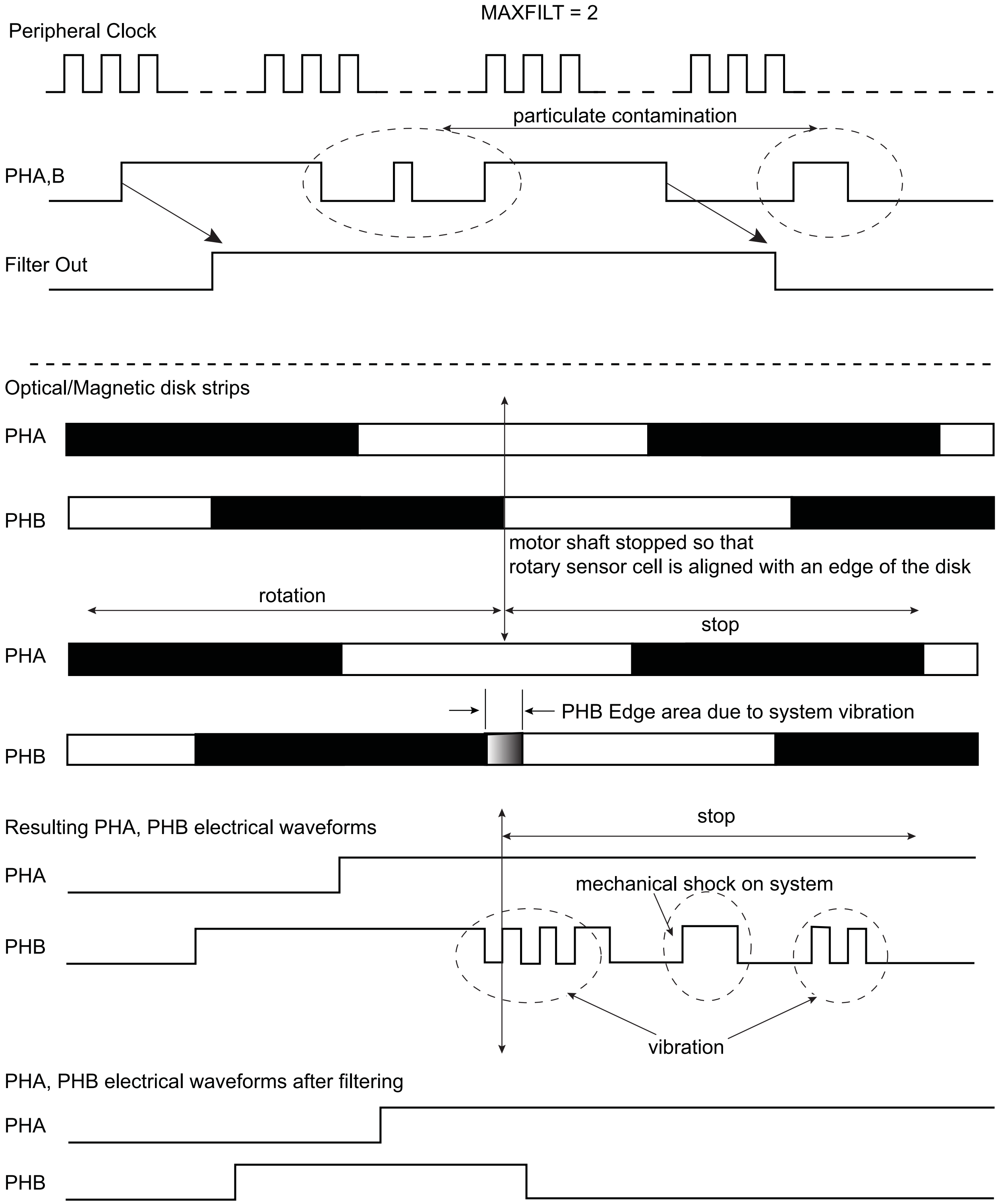
Figure 67-19. Input Stage



Input filtering can efficiently remove spurious pulses that might be generated by the presence of particulate contamination on the optical or magnetic disk of the rotary sensor.

Spurious pulses can also occur in environments with high levels of electromagnetic interference. Or, simply if vibration occurs even when rotation is fully stopped and the shaft of the motor is in such a position that the beginning of one of the reflective or magnetic bars on the rotary sensor disk is aligned with the light or magnetic (Hall) receiver cell of the rotary sensor. Any vibration can make the PHA, PHB signals toggle for a short duration.

Figure 67-20. Filtering Examples



67.6.16.3 Direction Status and Change Detection

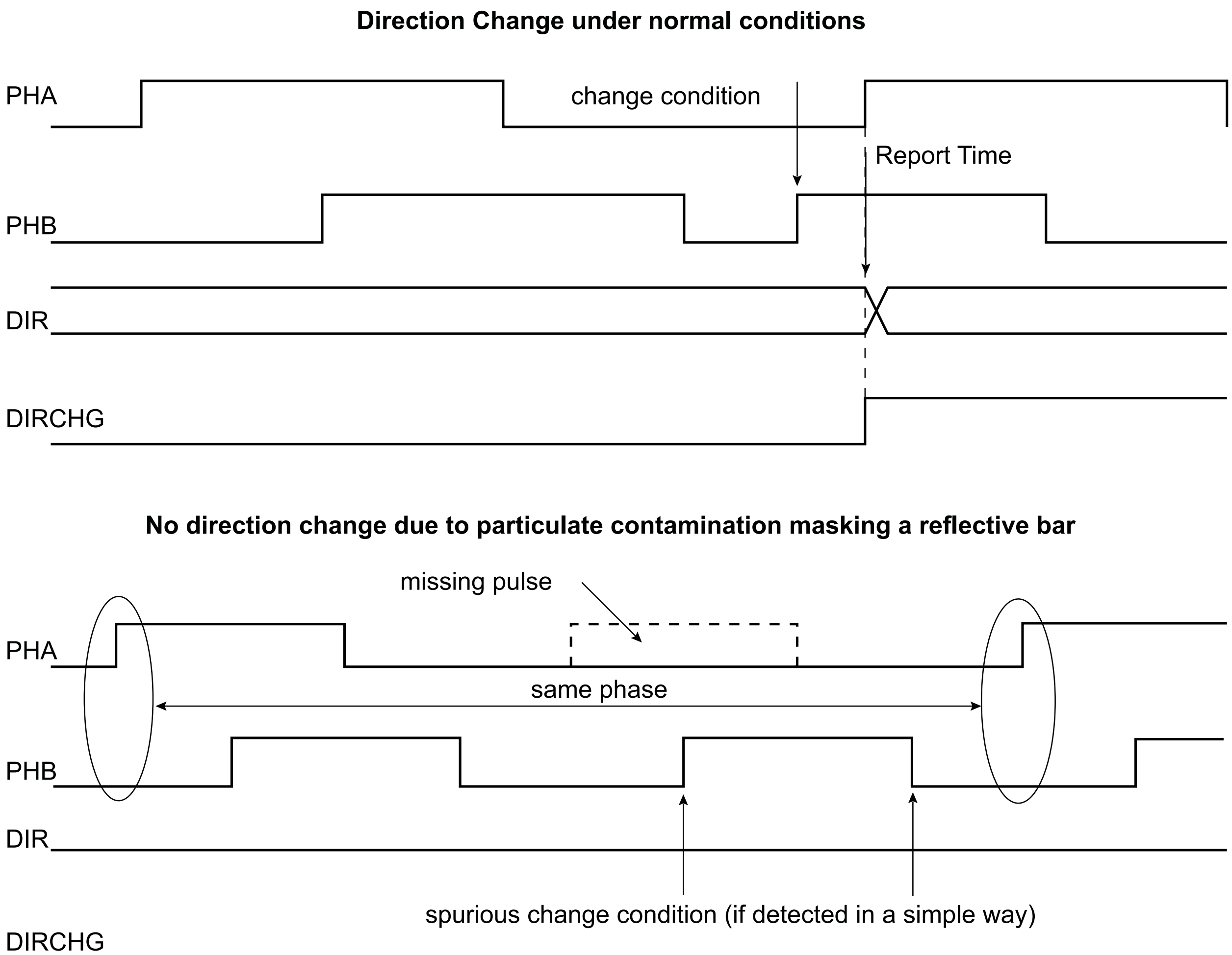
After filtering, the quadrature signals are analyzed to extract the rotation direction and edges of the two quadrature signals detected in order to be counted by TC logic downstream.

The direction status can be directly read at anytime in the TC_QISR. The polarity of the direction flag status depends on the configuration written in TC_BMR. INVA, INVB, INVIDX, SWAP modify the polarity of DIR flag.

Any change in rotation direction is reported in the TC_QISR and can generate an interrupt.

The direction change condition is reported as soon as two consecutive edges on a phase signal have sampled the same value on the other phase signal and there is an edge on the other signal. The two consecutive edges of one phase signal sampling the same value on other phase signal is not sufficient to declare a direction change, as particulate contamination may mask one or more reflective bars on the optical or magnetic disk of the sensor. See the following figure for waveforms.

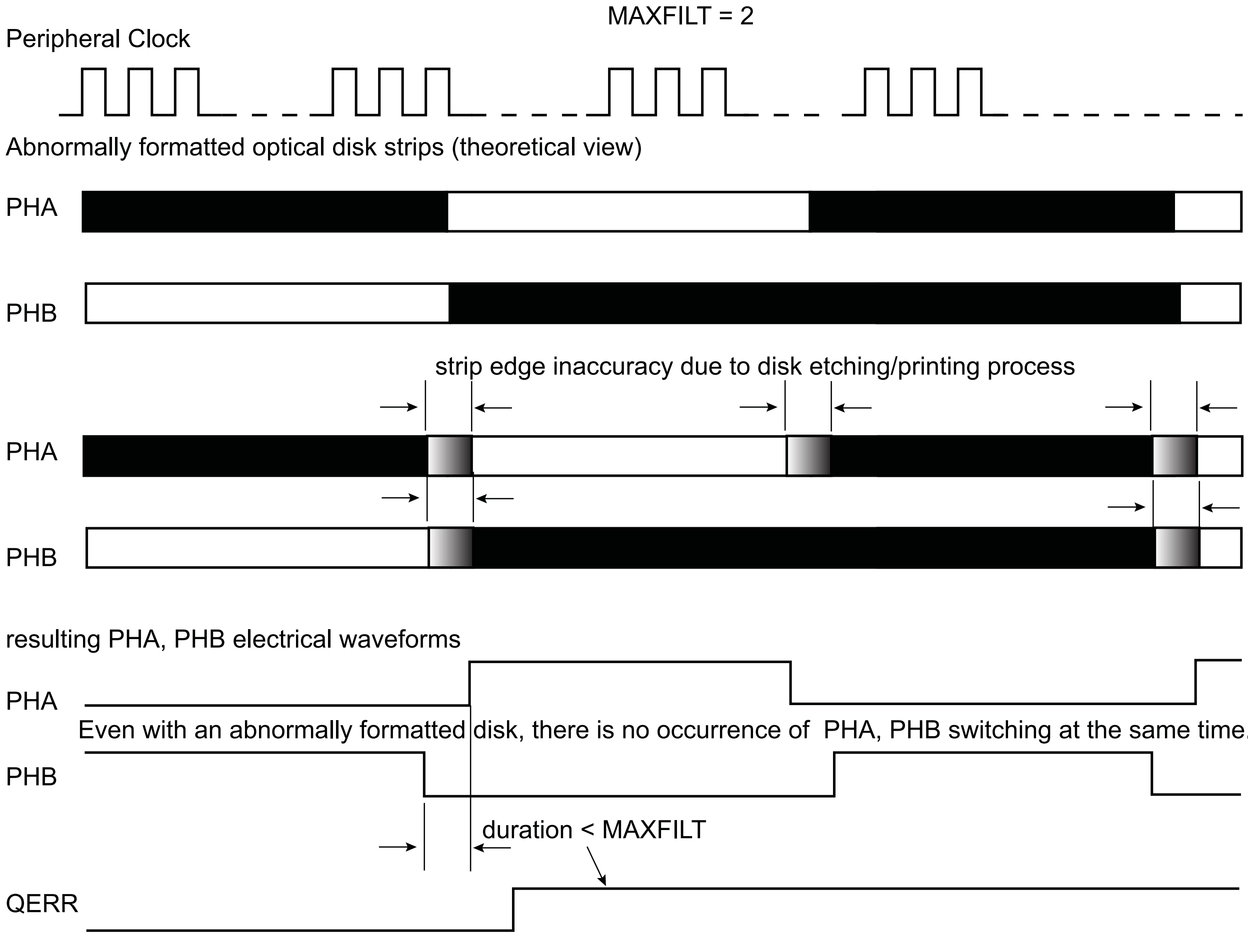
Figure 67-21. Rotation Change Detection



The direction change detection is disabled when TC_BMR.QDTRANS is set. In this case, the DIR flag report must not be used.

A quadrature error is also reported by the QDEC via TC_QISR.QERR. This error is reported if the time difference between two edges on PHA, PHB is lower than a predefined value. This predefined value is configurable and corresponds to (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns. After being filtered, there is no reason to have two edges closer than (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns under normal mode of operation.

Figure 67-22. Quadrature Error Detection



MAXFILT must be tuned according to several factors such as the peripheral clock frequency, type of rotary sensor and rotation speed to be achieved.

67.6.16.4 Position and Rotation Measurement

When TC_BMR.POSEN is set, the motor axis position is processed on channel 0 (by means of the PHA, PHB edge detections) and the number of motor revolutions are recorded on channel 1 if the IDX signal is provided on the TIOB1 input. If no IDX signal is available, the internal counter can be cleared for each revolution if the number of counts per revolution is configured in TC_RC0.RC and the TC_CMR.CPCTRG bit is written to ‘1’. The position measurement can be read in the TC_CV0 register and the rotation measurement can be read in the TC_CV1 register.

Channel 0 and 1 must be configured in Capture mode (TC_CMR0.WAVE = 0). ‘Rising edge’ must be selected as the External Trigger Edge (TC_CMR.ETRGEDG = 0x01) and ‘TIOAx’ must be selected as the External Trigger (TC_CMR.ABETRG = 0x1). The process must be started by configuring TC_CCR.CLKEN and TC_CCR.SWTRG.

In parallel, the number of edges are accumulated on TC channel 0 and can be read on the TC_CV0 register.

Therefore, the accurate position can be read on both TC_CV registers and concatenated to form a 32-bit word.

The TC channel 0 is cleared for each increment of IDX count value.

Depending on the quadrature signals, the direction is decoded and allows to count up or down in TC channels 0 and 1. The direction status is reported on TC_QISR.

67.6.16.5 Speed Measurement

When TC_BMR.SPEEDEN is set, the speed measure is enabled on channel 0.

A time base must be defined on channel 2 by writing the TC_RC2 period register. Channel 2 must be configured in Waveform mode (WAVE bit set) in TC_CMR2. The WAVSEL field must be defined with 0x10 to clear the counter by comparison and matching with TC_RC value. Field ACPC must be defined at 0x11 to toggle TIOAx output.

This time base is automatically fed back to TIOAx of channel 0 when QDEN and SPEEDEN are set.

Channel 0 must be configured in Capture mode (WAVE = 0 in TC_CMR0). TC_CMR0.ABETRG must be configured at 1 to select TIOAx as a trigger for this channel.

EDGTRG must be set to 0x01, to clear the counter on a rising edge of the TIOAx signal and field LDRA must be set accordingly to 0x01, to load TC_RA0 at the same time as the counter is cleared (LDRB must be set to 0x01). As a consequence, at the end of each time base period the differentiation required for the speed calculation is performed.

The process must be started by configuring bits CLKEN and SWTRG in the TC_CCR.

The speed can be read on field RA in TC_RA0.

Channel 1 can still be used to count the number of revolutions of the motor.

67.6.16.6 Detecting a Missing Index Pulse

To detect a missing index pulse due contamination, dust, etc., the TC_SR0.CPCS flag can be used. It is also possible to assert the interrupt line if the TC_SR0.CPCS flag is enabled as a source of the interrupt by writing a ‘1’ to TC_IER0.CPCS.

The TC_RC0.RC field must be written with the nominal number of counts per revolution provided by the rotary encoder, plus a margin to eliminate potential noise (e.g., if nominal count per revolution is 1024, then TC_RC0.RC=1026).

If the index pulse is missing, the timer value is not cleared and the nominal value is exceeded, then the comparator on the RC triggers an event, TC_SR0.CPCS=1, and the interrupt line is asserted if TC_IER0.CPCS=1.

The missing index pulse detection is only valid if the bit TC_QISR.DIRCHG=0.

67.6.16.7 Detecting a Badly Located Index

The digital filter reduces effects of dust, scratches or contamination on the index line. If the contamination creates a pulse surpassing the filter capacity (in particular at low speed), this can be interpreted as an index pulse, even if it is badly placed.

An on-the-fly detection of a badly-placed index is enabled when TC_BMR.BIDXCE=1. TC_RC0.RC must be written with the nominal number of counts per revolution provided by the rotary encoder + 2, e.g., if the nominal count per revolution is 1024, then TC_RC0.RC=1026. This margin of 2 eliminates potential noise.

If the pulse is processed while the current value of the counter (TC_CV0) is outside the value programmed in TC_RC0.RC ±2, the TC_SR0.LDRAS flag is written to ‘1’ and the location of the pulse is written in TC_RA0.

This detection circuitry does not prevent a rotary encoder integrity check before use.

67.6.16.8 Detecting Contamination/Dust at Rotary Encoder Low Speed

The contamination/dust that can be filtered when the rotary encoder speed is high may not be filtered at low speed, thus creating unsollicited direction change, etc.

At low speed, even a minor contamination may appear as a long pulse, and thus not filtered and processed as a standard quadrature encoder pulse.

This contamination can be detected by using the similar method as the missing index detection.

A contamination exists on a phase line if TC_SR.CPCS = 1 and TC_QISR.DIRCHG = 1 when there is no sollicited change of direction.

67.6.16.9 Missing Pulse Detection and Autocorrection

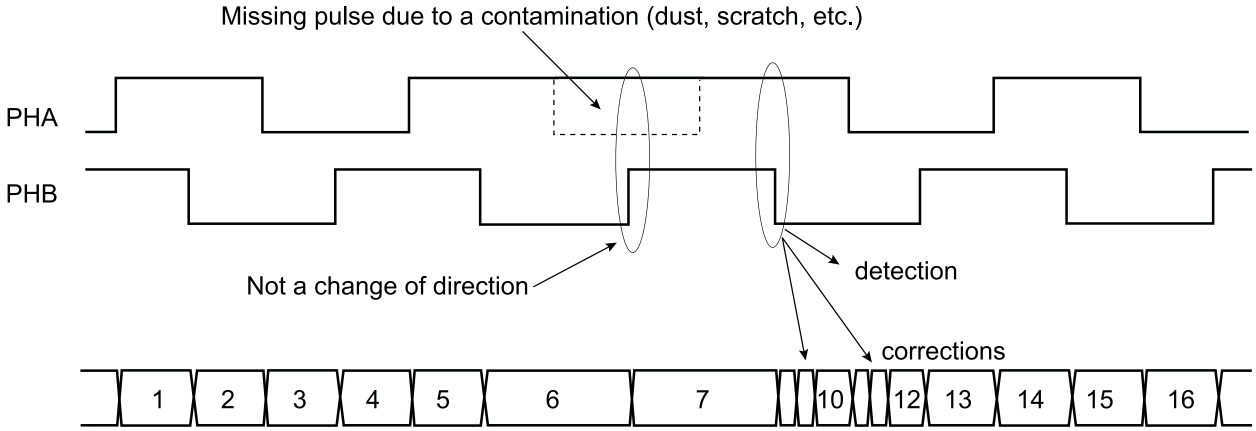
The QDEC is equipped with a circuitry which detects and corrects some errors that may result from contamination on optical disks or other materials producing the quadrature phase signals.

The detection and autocorrection only works if the Count mode is configured for both phases (EDGPHA = 1 in TC_BMR) and is enabled (AUTOC = 1 in TC_BMR).

If a pulse is missing on a phase signal, it is automatically detected and the pulse count reported in the CV field of the TC_CV0/1 is automatically corrected.

There is no detection if both phase signals are affected at the same location on the device providing the quadrature signals because the detection requires a valid phase signal to detect the contamination on the other phase signal.

Figure 67-23. Detection and Autocorrection of Missing Pulses



If a quadrature device is undamaged, the number of pulses counted for a predefined period of time must be the same with or without detection and autocorrection feature.

Therefore, if the measurement results differ, a contamination exists on the device producing the quadrature signals.

This does not substitute the measurements of the number of pulses between two index pulses (if available) but provides a complementary method to detect damaged quadrature devices.

When the device providing quadrature signals is severely damaged, potentially leading to a number of consecutive missing pulses greater than 1, the downstream processing may be affected. It is possible to define the maximum admissible number of consecutive missing pulses before issuing a Missing Pulse Error flag (MPE in TC_QISR). The threshold triggering an MPE flag report can be configured in TC_BMR.MAXCMP. If the field MAXCMP is cleared, MPE never rises. The flag MAXCMP can trigger an interrupt while the QDEC is operating, thus providing a real time report of a potential problem on the quadrature device.

67.6.16.10 Report of Filtered Pulses due to Contamination/Dust

When the digital filter removes pulses created by contamination/dust, a report is provided in the TC QDEC Interrupt Status Register. A separate flag is provided for each line (PHA, PHB, Index) and one flag is provided when a missing pulse is corrected (a ‘1’ must be written to TC_BMR.AUTOC). If TC_QISR.FPHA=1, a pulse has been filtered on PHA line. If TC_QISR.FPHB=1, a pulse has been filtered on PHB line. If TC_QISR.FIDX=1, a pulse has been filtered on Index line. If TC_QISR.FMP=1, a missing pulse has been corrected by the missing pulse detection logic.

The on-the-fly detection and associated flags can be used to anticipate further effects of contamination/dust, in particular if the flag is written to ‘1’ when the rotary encoder speed is high. This may cause abnormal behavior when the rotary encoder slows down if the abnormal pulse is no longer filtered (surpassing digital filter capabilities).

This detection circuitry does not prevent rotary encoder integrity check before use.

67.6.17 2-bit Gray Up/Down Counter for Stepper Motor

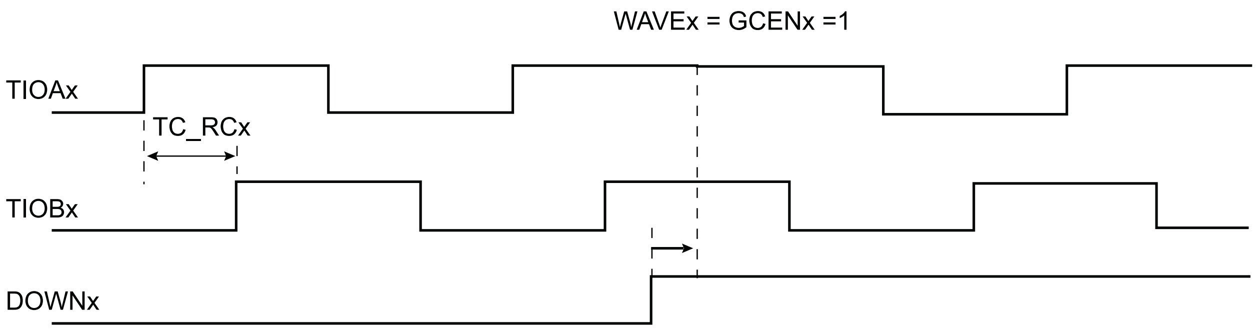
Each channel can be independently configured to generate a 2-bit Gray count waveform on corresponding TIOAx, TIOBx outputs by means of TC_SMMRx.GCEN.

Up or Down count can be defined by writing TC_SMMRx.DOWN.

It is mandatory to configure the channel in Waveform mode in the TC_CMR.

The period of the counters can be programmed in TC_RCx.

Figure 67-24. 2-bit Gray Up/Down Counter



67.6.18 Fault Mode

At any time, the TC_RCx registers can be used to perform a comparison on the respective current channel counter value (TC_CVx) with the value of TC_RCx register.

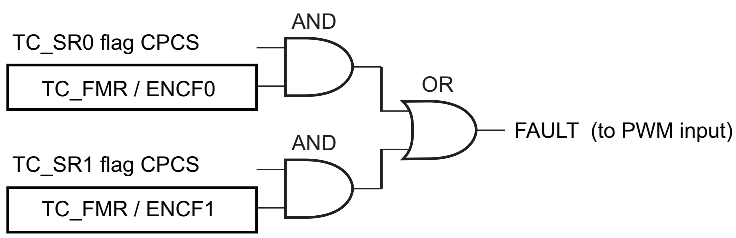
The CPCSx flags can be set accordingly and an interrupt can be generated.

This interrupt is processed but requires an unpredictable amount of time to be achieve the required action.

It is possible to trigger the FAULT output of the TIMER1 with CPCS from TC_SR0 and/or CPCS from TC_SR1. Each source can be independently enabled/disabled in the TC_FMR.

This can be useful to detect an overflow on speed and/or position when QDEC is processed and to act immediately by using the FAULT output.

Figure 67-25. Fault Output Generation



67.6.19 Register Write Protection

To prevent any single software error from corrupting TC behavior, certain registers in the address space can be write-protected by setting the WPEN bit , WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) in the TC Write Protection Mode Register (TC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the "TC Safety Status Register" (TC_SSRx) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The Timer Counter clock of the first channel must be enabled to access TC_WPMR.

The following registers can be write-protected when WPEN is set:

- TC Block Mode Register
- TC Channel Mode Register Capture Mode
- TC Channel Mode Register Waveform Mode
- TC Fault Mode Register
- TC Stepper Motor Mode Register
- TC Register A
- TC Register B
- TC Register C
- TC Extended Mode Register

The following registers can be write-protected when WPITEN is set:

- TC Interrupt Enable Register
- TC Interrupt Disable Register
- TC QDEC Interrupt Enable Register
- TC QDEC Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TC Channel Control Register
- TC Block Control Register

67.6.20 Security and Safety Analysis and Reports

Several type of checks are performed when a TC channel is enabled.

The peripheral clock of the TC is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TDES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TC_SSRx.CGD is set, an abnormal condition occurred on the internal clock net. This flag is not set under normal operating conditions.

The internal counter of a TC channel is also monitored and if an abnormal state is detected, the flag TC_SSRx.SEQE is set. This flag cannot be set under normal operating conditions.

The software accesses to the TC are monitored and if an incorrect access is performed, the flag TC_SSRx.SWE is set. The type of incorrect/abnormal software access is reported in TC_SSRx.SWETYP (see TC_CSRx for details). As an example, when the TC channel is configured in Capture mode, reading TC_RAx when TC_SRx.LDRAS=0 asserts the SWE flags. TC_SSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE, WPVS are automatically cleared when TC_SSRx is read.

If one of these flags is set, the flag TC_SRx.SECE is set and triggers an interrupt if the TC_IMRx.SECE bit is ‘1’. SECE is cleared by reading TC_SRx.

67.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TC_CCR0	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x04	TC_CMR0 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x04	TC_CMR0 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x08	TC_SMMR0	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x0C	TC_RAB0	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x10	TC_CV0	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x14	TC_RA0	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x18	TC_RB0	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x1C	TC_RC0	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x20	TC_SR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x24	TC_IER0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x28	TC_IDR0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x2C	TC_IMR0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x30	TC_EMR0	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	TC_CSR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x38	TC_SSR0	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x3C ... 0x3F	Reserved									
0x40	TC_CCR1	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x44	TC_CMR1 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x44	TC_CMR1 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x48	TC_SMMR1	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x4C	TC_RAB1	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x50	TC_CV1	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x54	TC_RA1	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x58	TC_RB1	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x5C	TC_RC1	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x60	TC_SR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x64	TC_IER1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x68	TC_IDR1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x6C	TC_IMR1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x70	TC_EMR1	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0x74	TC_CSR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x78	TC_SSR1	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x7C ... 0x7F	Reserved									
0x80	TC_CCR2	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x84	TC_CMR2 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x84	TC_CMR2 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x88	TC_SMMR2	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x8C	TC_RAB2	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x90	TC_CV2	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x94	TC_RA2	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x98	TC_RB2	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x9C	TC_RC2	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0xA0	TC_SR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA4	TC_IER2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xA8	TC_IDR2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xAC	TC_IMR2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xB0	TC_EMR2	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0xB4	TC_CSR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0xB8	TC_SSR2	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0xBC ... 0xBF	Reserved									
0xC0	TC_BCR	31:24								
		23:16								
		15:8								
		7:0								SYNC
0xC4	TC_BMR	31:24			MAXCMP[3:0]				MAXFILT[5:4]	
		23:16	MAXFILT[3:0]					AUTOC	IDXPHB	SWAP
		15:8	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
		7:0			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
0xC8	TC_QIER	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xCC	TC_QIDR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD0	TC_QIMR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD4	TC_QISR	31:24								
		23:16								
		15:8								DIR
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD8	TC_FMR	31:24								
		23:16								
		15:8								
		7:0							ENCF1	ENCF0
0xDC	TC_QSR	31:24								
		23:16								
		15:8								DIR
		7:0								

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0xE0 ... 0xE3	Reserved									
0xE4	TC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN

67.7.1 TC Channel Control Register

Name: TC_CCRx
Offset: 0x00 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SWTRG	CLKDIS	CLKEN
Access						W	W	W
Reset						–	–	–

Bit 2 – SWTRG Software Trigger Command

Value	Description
0	No effect.
1	A software trigger is performed: the counter is reset and the clock is started.

Bit 1 – CLKDIS Counter Clock Disable Command

Value	Description
0	No effect.
1	Disables the clock.

Bit 0 – CLKEN Counter Clock Enable Command

Value	Description
0	No effect.
1	Enables the clock if CLKDIS is not 1.

67.7.2 TC Channel Mode Register: Capture Mode

Name: TC_CMRx (CAPTURE MODE)
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can be written only if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 22:20 – SBSMPLR[2:0] Loading Edge Subsampling Ratio

Value	Name	Description
0	ONE	Load a Capture register each selected edge.
1	HALF	Load a Capture register every 2 selected edges.
2	FOURTH	Load a Capture register every 4 selected edges.
3	EIGHTH	Load a Capture register every 8 selected edges.
4	SIXTEENTH	Load a Capture register every 16 selected edges.

Bits 19:18 – LDRB[1:0] RB Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bits 17:16 – LDRA[1:0] RA Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bit 15 – WAVE Waveform Mode

Value	Description
0	Capture mode is enabled.
1	Capture mode is disabled (Waveform mode is enabled).

Bit 14 – CPCTRG RC Compare Trigger Enable

Value	Description
0	RC Compare has no effect on the counter and its clock.
1	RC Compare resets the counter and starts the counter clock.

Bit 10 – ABETRG TIOAx or TIOBx External Trigger Selection

Value	Description
0	TIOBx is used as an external trigger.
1	TIOAx is used as an external trigger.

Bits 9:8 – ETRGEDG[1:0] External Trigger Edge Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – LDBDIS Counter Clock Disable with RB Loading

Value	Description
0	Counter clock is not disabled when RB loading occurs.
1	Counter clock is disabled when RB loading occurs.

Bit 6 – LDBSTOP Counter Clock Stopped with RB Loading

Value	Description
0	Counter clock is not stopped when RB loading occurs.
1	Counter clock is stopped when RB loading occurs.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection
To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [TC_ID] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal TD_SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

67.7.3 TC Channel Mode Register: Waveform Mode

Name: TC_CMRx (WAVEFORM MODE)
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – BSWTRG[1:0] Software Trigger Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 29:28 – BEEVT[1:0] External Event Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 27:26 – BCPC[1:0] RC Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 25:24 – BCPB[1:0] RB Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 23:22 – ASWTRG[1:0] Software Trigger Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 21:20 – AEEVT[1:0] External Event Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 19:18 – ACPC[1:0] RC Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 17:16 – ACPA[1:0] RA Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bit 15 – WAVE Waveform Mode

Value	Description
0	Waveform mode is disabled (Capture mode is enabled).
1	Waveform mode is enabled.

Bits 14:13 – WAVSEL[1:0] Waveform Selection

Value	Name	Description
0	UP	UP mode without automatic trigger on RC Compare
1	UPDOWN	UPDOWN mode without automatic trigger on RC Compare
2	UP_RC	UP mode with automatic trigger on RC Compare
3	UPDOWN_RC	UPDOWN mode with automatic trigger on RC Compare

Bit 12 – ENETRG External Event Trigger Enable
Whatever the value programmed in ENETRG, the selected external event only controls the TIOAx output and TIOBx if not used as input (trigger event input or other input used).

Value	Description
0	The external event has no effect on the counter and its clock.
1	The external event resets the counter and starts the counter clock.

Bits 11:10 – EEVT[1:0] External Event Selection
Signal selected as external event.

Value	Name	Description	TIOB Direction
0	TIOB	TIOB	Input
1	XC0	XC0	Output
2	XC1	XC1	Output
3	XC2	XC2	Output

Note: If TIOB is chosen as the external event signal, it is configured as an input and no longer generates waveforms and subsequently no IRQs.

Bits 9:8 – EEVTEDG[1:0] External Event Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – CPCDIS Counter Clock Disable with RC Compare

Value	Description
0	Counter clock is not disabled when counter reaches RC.
1	Counter clock is disabled when counter reaches RC.

Bit 6 – CPCSTOP Counter Clock Stopped with RC Compare

Value	Description
0	Counter clock is not stopped when counter reaches RC.
1	Counter clock is stopped when counter reaches RC.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection
To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [TC_ID] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal TD_SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

67.7.4 TC Stepper Motor Mode Register

Name: TC_SMMRx
Offset: 0x08 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DOWN	GCEN
Access							R/W	R/W
Reset							0	0

Bit 1 – DOWN Down Count

Value	Description
0	Up counter.
1	Down counter.

Bit 0 – GCEN Gray Count Enable

Value	Description
0	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by internal counter of channel x.
1	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by a 2-bit Gray counter.

67.7.5 TC Register AB

Name: TC_RABx
Offset: 0x0C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RAB[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RAB[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RAB[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RAB[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RAB[31:0] Register A or Register B
RAB contains the next unread capture Register A or Register B value in real time. It is usually read by the DMA after a request due to a valid load edge on TIOAx.
When DMA is used, the RAB register address must be configured as source address of the transfer.

67.7.6 TC Counter Value Register

Name: TC_CVx
Offset: 0x10 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CV[31:0] Counter Value
CV contains the counter value in real time.

Important:
For 16-bit channels, the CV field size is limited to register bits 15:0.

67.7.7 TC Register A

Name: TC_RAx
Offset: 0x14 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RA[31:0] Register A
RA contains the Register A value in real time.

67.7.8 TC Register B

Name: TC_RBx
Offset: 0x18 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RB[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RB[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RB[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RB[31:0] Register B
RB contains the Register B value in real time.

67.7.9 TC Register C

Name: TC_RCx
Offset: 0x1C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RC[31:0] Register C
RC contains the Register C value in real time.

67.7.10 TC Interrupt Status Register

Name: TC_SRx
Offset: 0x20 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
								SECE
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx pin is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx pin is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx pin is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx pin is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	The clock is disabled.
1	The clock is enabled.

Bit 8 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred.
1	One or more safety or security event occurred since the last read of TC_SRx. For details on the event, see TC_SSRx.

Bit 7 – ETRGS External Trigger Status (cleared on read)

Value	Description
0	External trigger has not occurred since the last read of the Status register.
1	External trigger has occurred since the last read of the Status register.

Bit 6 – LDRBS RB Loading Status (cleared on read)

Value	Description
0	RB Load has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RB Load has occurred since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 5 – LDRAS RA Loading Status (cleared on read)

Value	Description
0	RA Load has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RA Load has occurred since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 4 – CPCS RC Compare Status (cleared on read)

Value	Description
0	RC Compare has not occurred since the last read of the Status register.
1	RC Compare has occurred since the last read of the Status register.

Bit 3 – CPBS RB Compare Status (cleared on read)

Value	Description
0	RB Compare has not occurred since the last read of the Status register or TC_CMRx.WAVE = 0.
1	RB Compare has occurred since the last read of the Status register, if TC_CMRx.WAVE = 1.

Bit 2 – CPAS RA Compare Status (cleared on read)

Value	Description
0	RA Compare has not occurred since the last read of the Status register or TC_CMRx.WAVE = 0.
1	RA Compare has occurred since the last read of the Status register, if TC_CMRx.WAVE = 1.

Bit 1 – LOVRS Load Overrun Status (cleared on read)

Value	Description
0	Load overrun has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 0 – COVFS Counter Overflow Status (cleared on read)

Value	Description
0	No counter overflow has occurred since the last read of the Status register.
1	A counter overflow has occurred since the last read of the Status register.

67.7.11 TC Interrupt Enable Register

Name: TC_IERx
Offset: 0x24 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Enable

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

67.7.12 TC Interrupt Disable Register

Name: TC_IDRx
Offset: 0x28 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Disable

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

67.7.13 TC Interrupt Mask Register

Name: TC_IMRx
Offset: 0x2C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						SECE		
Access						R		
Reset						0		
Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SECE Security and/or Safety Event Interrupt Mask

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

67.7.14 TC Extended Mode Register

Name: TC_EMRx
Offset: 0x30 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								NODIVCLK
Access								R/W
Reset								0
Bit	7	6	5	4	3	2	1	0
			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 8 – NODIVCLK No Divided Clock

Value	Description
0	The selected clock is defined by field TCCLKS in TC_CMRx.
1	The selected clock is peripheral clock and TCCLKS field (TC_CMRx) has no effect.

Bits 5:4 – TRIGSRCB[1:0] Trigger Source for Input B

Value	Name	Description
0	EXTERNAL_TIOBx	The trigger/capture input B is driven by external pin TIOBx
1	PWMx	For all channels: The trigger/capture input B is driven internally by the comparator output (see Synchronization with PWM) of the PWMx.

Bits 1:0 – TRIGSRCA[1:0] Trigger Source for Input A

Value	Name	Description
0	EXTERNAL_TIOAx	The trigger/capture input A is driven by external pin TIOAx
1	PWMx	For TC0, TC1.TIOB0, TC1.TIOB2: The trigger/capture input A is driven internally by PWMx. For TC1.TIOB1: The trigger/capture input B is driven internally by the GTSUCOMP signal of the Ethernet MAC (GMAC).

67.7.15 TC Channel Status Register

Name: TC_CSRx
Offset: 0x34 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Note: The flags in this register are a copy of the similar flags in the TC_SRx register. Reading the TC_CSRx does not perform a clear-on-read of TC_SRx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	Clock is disabled.
1	Clock is enabled.

67.7.16 TC Safety Status Register

Name: TC_SSRx
Offset: 0x38 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class

Value	Name	Description
0	WARNING	An abnormal access that does not have any impact.
1	ERROR	An abnormal access that may have an impact.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TC Channel x is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TC Channel x is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address of the TC (Warning).
3	W_RARB_CAPT	TC_RAx or TC_RBx are written while channel is enabled and configured in capture mode (Error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TC_SSRx.
1	A software error has occurred since the last read of TC_SSRx. The field SWETYP details the type of software error encountered.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No internal counter error has occurred since the last read of TC_SSRx. In normal operating conditions, SEQE is cleared.

Value	Description
1	An internal counter error has occurred since the last read of TC_SSRx. This flag can be set only under abnormal operating conditions resulting in clock glitch, etc. The detection is enabled if TC_CSRx.CLKSTA=1, TC_CMRx.WAVE=1, TC_CMRx.CPCTRG=1 and flag is set if TC_CVx > TC_RCx.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring has not been corrupted since the last read of TC_SSRx.
1	The clock monitoring has been corrupted since the last read of TC_SSRx. This flag can be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TC_SSRx.
1	A write protection violation has occurred since the last read of TC_SSRx. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

67.7.17 TC Block Control Register

Name: TC_BCR
Offset: 0xC0
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SYNC
Access								W
Reset								–

Bit 0 – SYNC Synchro Command

Value	Description
0	No effect.
1	Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

67.7.18 TC Block Mode Register

Name: TC_BMR
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			MAXCMP[3:0]				MAXFILT[5:4]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MAXFILT[3:0]					AUTOC	IDXPHB	SWAP
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:26 – MAXCMP[3:0] Maximum Consecutive Missing Pulses

Value	Description
0	The flag MPE in TC_QISR never rises.
1-15	Defines the number of consecutive missing pulses before a flag report.

Bits 25:20 – MAXFILT[5:0] Maximum Filter
Pulses with a period shorter than MAXFILT+1 peripheral clock cycles are discarded. For more details on MAXFILT constraints, see Input Preprocessing.

Value	Description
1-63	Defines the filtering capabilities.

Bit 18 – AUTOC AutoCorrection of missing pulses
0 (DISABLED): The detection and autocorrection function is disabled.
1 (ENABLED): The detection and autocorrection function is enabled.

Bit 17 – IDXPHB Index Pin is PHB Pin

Value	Description
0	IDX pin of the rotary sensor must drive TIOA1.
1	IDX pin of the rotary sensor must drive TIOB0.

Bit 16 – SWAP Swap PHA and PHB

Value	Description
0	No swap between PHA and PHB.
1	Swap PHA and PHB internally, prior to driving the QDEC.

Bit 15 – INVIDX Inverted Index

Value	Description
0	IDX (TIOA1) is directly driving the QDEC.

Value	Description
1	IDX is inverted before driving the QDEC.

Bit 14 – INVB Inverted PHB

Value	Description
0	PHB (TIOB0) is directly driving the QDEC.
1	PHB is inverted before driving the QDEC.

Bit 13 – INVA Inverted PHA

Value	Description
0	PHA (TIOA0) is directly driving the QDEC.
1	PHA is inverted before driving the QDEC.

Bit 12 – EDGPHA Edge on PHA Count Mode

Value	Description
0	Edges are detected on PHA only.
1	Edges are detected on both PHA and PHB.

Bit 11 – QDTRANS Quadrature Decoding Transparent

Value	Description
0	Full quadrature decoding logic is active (direction change detected).
1	Quadrature decoding logic is inactive (direction change inactive) but input filtering and edge detection are performed.

Bit 10 – SPEEDEN Speed Enabled

Value	Description
0	Disabled.
1	Enables the speed measure on channel 0, the time base being provided by channel 2.

Bit 9 – POSEN Position Enabled

Value	Description
0	Disable position.
1	Enables the position measure on channel 0 and 1.

Bit 8 – QDEN Quadrature Decoder Enabled
Quadrature decoding (direction change) can be disabled using QDTRANS bit.
One of the POSEN or SPEEDEN bits must be also enabled.

Value	Description
0	Disabled.
1	Enables the QDEC (filter, edge detection and quadrature decoding).

Bits 5:4 – TC2XC2S[1:0] External Clock Signal 2 Selection

Value	Name	Description
0	TCLK2	Signal connected to XC2: TCLK2
1	–	Reserved
2	TIOA0	Signal connected to XC2: TIOA0
3	TIOA1	Signal connected to XC2: TIOA1

Bits 3:2 – TC1XC1S[1:0] External Clock Signal 1 Selection

Value	Name	Description
0	TCLK1	Signal connected to XC1: TCLK1
1	–	Reserved
2	TIOA0	Signal connected to XC1: TIOA0
3	TIOA2	Signal connected to XC1: TIOA2

Bits 1:0 – TC0XC0S[1:0] External Clock Signal 0 Selection

Value	Name	Description
0	TCLK0	Signal connected to XC0: TCLK0
1	–	Reserved
2	TIOA1	Signal connected to XC0: TIOA1
3	TIOA2	Signal connected to XC0: TIOA2

67.7.19 TC QDEC Interrupt Enable Register

Name: TC_QIER
Offset: 0xC8
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Enables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Enables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Enables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Enables the interrupt when phase A line has a filtered contamination.

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	No effect.
1	Enables the interrupt when an occurrence of MAXCMP consecutive missing pulses is detected.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.

Value	Description
1	Enables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.
1	Enables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Enables the interrupt when a rising edge occurs on IDX input.

67.7.20 TC QDEC Interrupt Disable Register

Name: TC_QIDR
Offset: 0xCC
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Disables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Disables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Disables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Disables the interrupt when phase A line has a filtered contamination.

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	No effect.
1	Disables the interrupt when an occurrence of MAXCMP consecutive missing pulses has been detected.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Disables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.
1	Disables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Disables the interrupt when a rising edge occurs on IDX input.

67.7.21 TC QDEC Interrupt Mask Register

Name: TC_QIMR
Offset: 0xD0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	The interrupt on auto-corrected missing pulse is disabled.
1	The interrupt on auto-corrected missing pulse is enabled.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	The interrupt on index line filtered contamination is disabled.
1	The interrupt on index line filtered contamination is enabled.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	The interrupt on phase B line filtered contamination is disabled.
1	The interrupt on phase B line filtered contamination is enabled.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	The interrupt on phase A line filtered contamination is disabled.
1	The interrupt on phase A line filtered contamination is enabled.

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is disabled.
1	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is enabled.

Bit 2 – QERR Quadrature Error

Value	Description
0	The interrupt on quadrature error is disabled.

Value	Description
1	The interrupt on quadrature error is enabled.

Bit 1 – DIRCHG Direction Change

Value	Description
0	The interrupt on rotation direction change is disabled.
1	The interrupt on rotation direction change is enabled.

Bit 0 – IDX Index

Value	Description
0	The interrupt on IDX input is disabled.
1	The interrupt on IDX input is enabled.

67.7.22 TC QDEC Interrupt Status Register

Name: TC_QISR
Offset: 0xD4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DIR
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No correction of missing pulse on phase A or B lines occurred since the last read of TC_QISR.
1	A correction of missing pulse on phase A or B lines occurred since the last read of TC_QISR.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No filtered contamination on index line since the last read of TC_QISR.
1	A contamination has been successfully on index line since the last read of TC_QISR.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No filtered contamination on phase B line since the last read of TC_QISR.
1	A contamination has been successfully on phase B line since the last read of TC_QISR.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No filtered contamination on phase A line since the last read of TC_QISR.
1	A contamination has been successfully on phase A line since the last read of TC_QISR.

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	The number of consecutive missing pulses has not reached the maximum value specified in MAXCMP since the last read of TC_QISR.
1	An occurrence of MAXCMP consecutive missing pulses has been detected since the last read of TC_QISR.

Bit 2 – QERR Quadrature Error

Value	Description
0	No quadrature error since the last read of TC_QISR.
1	A quadrature error occurred since the last read of TC_QISR.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No change on rotation direction since the last read of TC_QISR.
1	The rotation direction changed since the last read of TC_QISR.

Bit 0 – IDX Index

Value	Description
0	No Index input change since the last read of TC_QISR.
1	The IDX input has changed since the last read of TC_QISR.

67.7.23 TC Fault Mode Register

Name: TC_FMR
Offset: 0xD8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENCF1	ENCF0
Access							R/W	R/W
Reset							0	0

Bit 1 – ENCF1 Enable Compare Fault Channel 1

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 1.
1	Enables the FAULT output source (CPCS flag) from channel 1.

Bit 0 – ENCF0 Enable Compare Fault Channel 0

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 0.
1	Enables the FAULT output source (CPCS flag) from channel 0.

67.7.24 TC QDEC Status Register

Name: TC_QSR
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Note: The flag in this register is a copy of the similar flag in the TC_QISR register. Reading the TC_QSR does not perform a clear-on-read of TC_QISR flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DIR
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

67.7.25 TC Write Protection Mode Register

Name: TC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
`0x54494D`	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
`0`	The last write protection violation source is reported in TC_SSRx.WPVSRC and the last software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.
`1`	Only the first write protection violation source is reported in TC_SSRx.WPVSRC and only the first software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
`0`	Disables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
`1`	Enables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
`0`	Disables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
`1`	Enables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 0 – WPEN Write Protection Enable
The Timer Counter clock of the first channel must be enabled to access this register.
See Register Write Protection for a list of registers that can be write-protected and Timer Counter clock conditions.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

68. Pulse Width Modulation Controller (PWM)

68.1 Description

The Pulse Width Modulation Controller (PWM) generates output pulses on 4 channels independently according to parameters defined per channel. Each channel controls two complementary square output waveforms. Characteristics of the output waveforms such as period, duty-cycle, polarity and dead-times (also called dead-bands or non-overlapping times) are configured through the user interface. Each channel selects and uses one of the clocks provided by the clock generator. The clock generator provides several clocks resulting from the division of the PWM peripheral clock. External triggers can modify the output values in real time.

All accesses to the PWM are made through registers mapped on the peripheral bus. All channels integrate a double buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum, the duty-cycle or the dead-times.

Channels can be linked together as synchronous channels to be able to update their duty-cycle or dead-times at the same time.

The update of duty-cycles of synchronous channels can be performed by the DMA Controller channel which offers buffer transfer without processor Intervention.

The PWM includes a spread-spectrum counter to allow a constantly varying period (only for Channel 0). This counter may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

The PWM provides 8 independent comparison units capable of comparing a programmed value to the counter of the synchronous channels (counter of channel 0). These comparisons are intended to generate software interrupts, to trigger pulses on the 2 independent event lines (in order to synchronize ADC conversions with a lot of flexibility independently of the PWM outputs) and to trigger DMA Controller transfer requests.

PWM outputs can be overridden synchronously or asynchronously to their channel counter.

The PWM provides a fault protection mechanism with 6 fault inputs, capable to detect a fault condition and to override the PWM outputs asynchronously (outputs forced to ‘0’, ‘1’ or Hi-Z).

For safety usage, some configuration registers are write-protected.

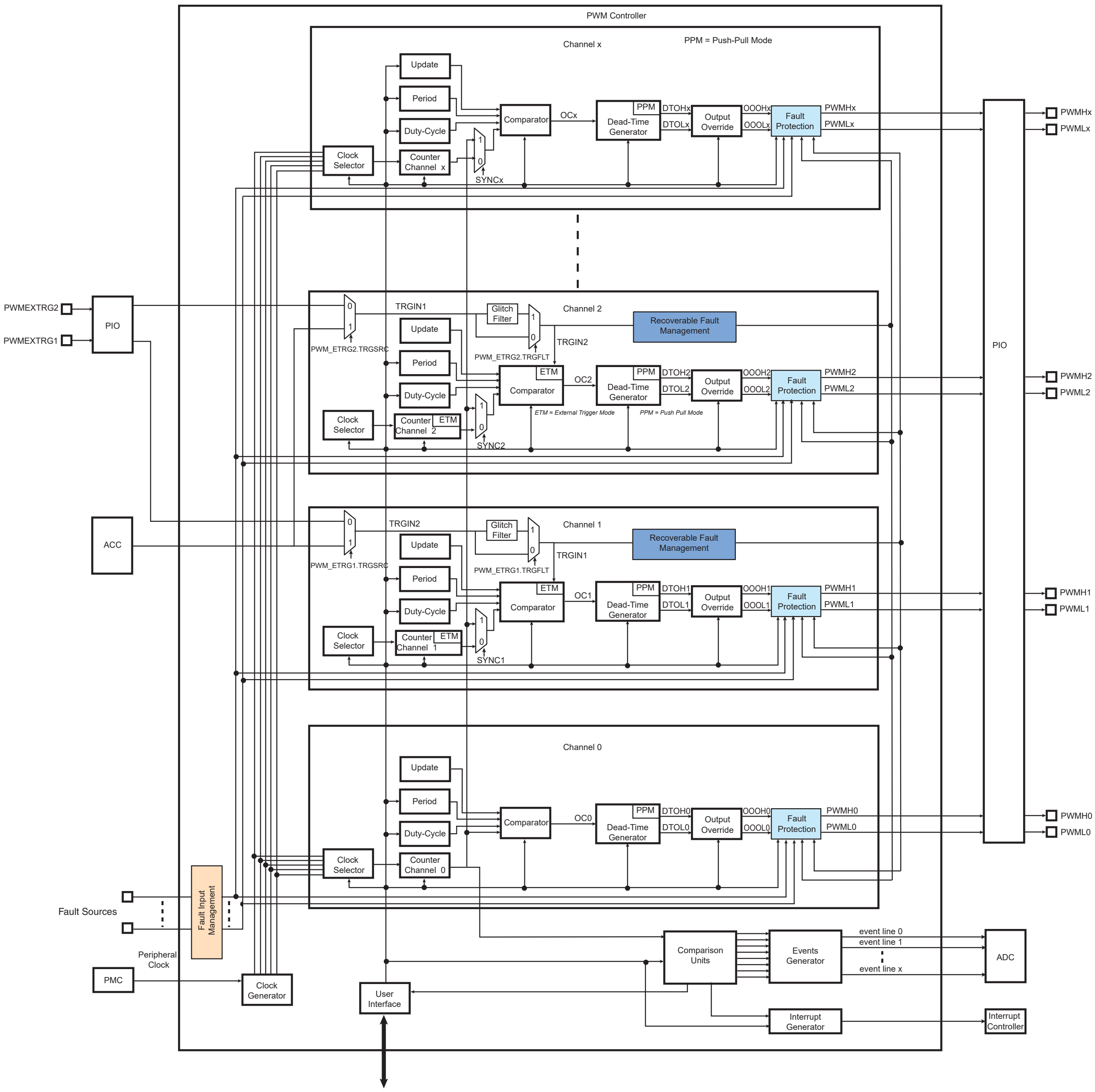
68.2 Embedded Characteristics

- 4 Channels
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n counter providing eleven clocks
  - Two independent linear dividers working on Modulo n counter outputs
- Independent Channels
  - Independent 16-bit counter for each channel
  - Independent complementary outputs with 12-bit dead-time generator (also called dead-band or non-overlapping time) for each channel
  - Independent Push-Pull mode for each channel
  - Independent enable-disable command for each channel
  - Independent clock selection for each channel
  - Independent period, duty-cycle and dead-time for each channel
  - Independent double buffering of period, duty-cycle and dead-times for each channel
  - Independent programmable selection of the output waveform polarity for each channel, with double buffering
  - Independent programmable center- or left-aligned output waveform for each channel
  - Independent output override for each channel
  - Independent interrupt for each channel, at each period for left-aligned or center-aligned configuration

  - Independent update time selection of double buffering registers (polarity, duty cycle) for each channel, at each period for left-aligned or center-aligned configuration
- External Trigger Input Management (e.g., for DC/DC or Lighting Control)
  - External PWM Reset mode
  - External PWM Start mode
  - Cycle-by-cycle duty cycle mode
  - Leading-edge blanking
- 2 2-bit Gray Up/Down Channels for Stepper Motor Control
- Spread Spectrum Counter to Allow a Constantly Varying Duty Cycle (only for Channel 0)
- Synchronous Channel Mode
  - Synchronous channels share the same counter
  - Mode to update the synchronous channels registers after a programmable number of periods
  - Synchronous channels support connection of one DMA Controller channel offers buffer transfer without processor intervention to update duty-cycle registers
- 2 Independent Event Lines Intended to Synchronize ADC Conversions
  - Programmable delay for event lines to delay ADC measurements
- 8 Comparison Units Intended to Generate Interrupts, Pulses on Event Lines and DMA Controller Transfer Requests
- 6 Programmable Fault Inputs Providing Asynchronous Protection of PWM Outputs
  - Two driven by the user through PIO inputs
  - Driven by the PMC when crystal oscillator clock fails
  - Driven by the ADC Controller through configurable comparison function
  - Driven by the Timer/Counter through configurable comparison function
- Register Write Protection

68.3 Block Diagram

Figure 68-1. PWM Controller Block Diagram



68.4 I/O Lines Description

Each channel outputs two complementary external I/O lines.

Table 68-1. I/O Lines Description

Name	Description	Type
PWMHx	PWM Waveform Output High for channel x	Output
PWMLx	PWM Waveform Output Low for channel x	Output
PWMFIx	PWM Fault Input x	Input
PWMEXTRGy	PWM Trigger Input y	Input

68.5 Product Dependencies

68.5.1 I/O Lines

The pins used for interfacing the PWM are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the desired PWM pins to their peripheral function. If I/O lines of the PWM are not used by the application, they can be used for other purposes by the PIO controller.

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines are assigned to PWM outputs.

68.5.2 Power Management

The PWM is not continuously clocked. The programmer must first enable the PWM clock in the Power Management Controller (PMC) before using the PWM. However, if the application does not require PWM operations, the PWM clock can be stopped when not needed and be restarted later. In this case, the PWM will resume its operations where it left off.

68.5.3 Interrupt Sources

The PWM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the PWM interrupt requires the Interrupt Controller to be programmed first.

68.5.4 Fault Inputs

The PWM has the fault inputs connected to the different modules. Refer to the implementation of these modules within the product for detailed information about the fault generation procedure. The PWM receives faults from:

- PIO inputs
- the PMC
- the ADC controller
- Timer/Counters

Table 68-2. Fault Inputs

Fault Generator	External PWM Fault Input Number	Polarity Level[(1)]	Fault Input ID
PA7/PB19/PB28/PD24/PE0	PWMFI0	User-defined	0
PA8/PB20/PB29/PD25/PE1	PWMFI1	User-defined	1
PMC	–	To be configured to 1	2
ADC	–	To be configured to 1	3
Timer0	–	To be configured to 1	4

Note:
1. FPOL field in PWM_FMR.

68.5.5 External Trigger Inputs

Table 68-3. External Trigger Inputs

IO Pin Name	External Trigger Pin Name	Polarity Level
PA5/PB17/PB26/PD22/PD30	PWMEXTRG0	1
PA6/PB18/PB27/PD23/PD31	PWMEXTRG1	1

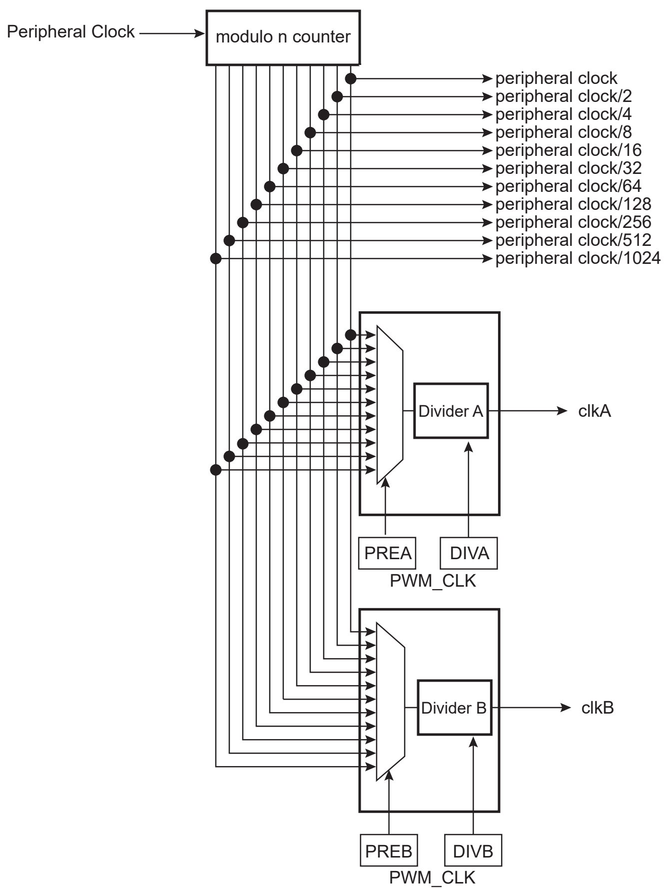
68.6 Functional Description

The PWM controller is primarily composed of a clock generator module and 4 channels.

- Clocked by the peripheral clock, the clock generator module provides 13 clocks.
- Each channel can independently choose one of the clock generator outputs.
- Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

68.6.1 PWM Clock Generator

Figure 68-2. Functional View of the Clock Generator Block Diagram



The PWM peripheral clock is divided in the clock generator module to provide different clocks available for all channels. Each channel can independently select one of the divided clocks.

The clock generator is divided into different blocks:

- a modulo n counter which provides 11 clocks:
  - $f_{peripheral\ clock}$
  - $f_{peripheral\ clock}/2$

– $f_{peripheral\ clock}/4$
– $f_{peripheral\ clock}/8$
– $f_{peripheral\ clock}/16$
– $f_{peripheral\ clock}/32$
– $f_{peripheral\ clock}/64$
– $f_{peripheral\ clock}/128$
– $f_{peripheral\ clock}/256$
– $f_{peripheral\ clock}/512$
– $f_{peripheral\ clock}/1024$

- two linear dividers (1, 1/2, 1/3, ... 1/255) that provide two separate clocks: clkA and clkB

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Clock register (PWM_CLK). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value.

After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) are set to ‘0’. This implies that after reset clkA (clkB) are turned off.

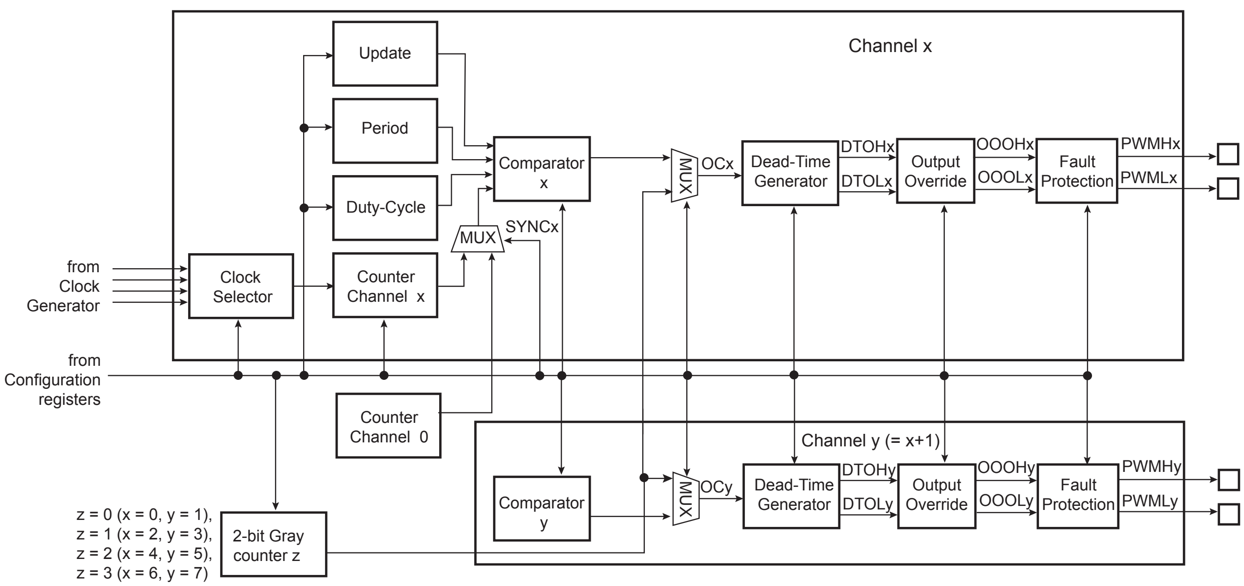
At reset, all clocks provided by the modulo n counter are turned off except the peripheral clock. This situation is also true when the PWM peripheral clock is turned off through the Power Management Controller.

⚠ CAUTION Before using the PWM controller, the programmer must first enable the peripheral clock in the Power Management Controller (PMC).

68.6.2 PWM Channel

68.6.2.1 Channel Block Diagram

Figure 68-3. Functional View of the Channel Block Diagram



Each of the 4 channels is composed of six blocks:

- A clock selector which selects one of the clocks provided by the clock generator (described in PWM Clock Generator).
- A counter clocked by the output of the clock selector. This counter is incremented or decremented according to the channel configuration and comparators matches. The size of the counter is 16 bits.

- A comparator used to compute the OCx output waveform according to the counter value and the configuration. The counter value can be the one of the channel counter or the one of the channel 0 counter according to SYNCx bit in the PWM Sync Channels Mode Register (PWM_SCM).
- A 2-bit configurable Gray counter enables the stepper motor driver. One Gray counter drives 2 channels.
- A dead-time generator providing two complementary outputs (DTOHx/DTOLx) which allows to drive external power control switches safely.
- An output override block that can force the two complementary outputs to a programmed value (OOOHx/OOOLx).
- An asynchronous fault protection mechanism that has the highest priority to override the two complementary outputs (PWMHx/PWMLx) in case of fault detection (outputs forced to ‘0’, ‘1’ or Hi-Z).

68.6.2.2 Comparator

The comparator continuously compares its counter value with the channel period defined by CPRD in the PWM Channel Period Register (PWM_CPRDx) and the duty-cycle defined by CDTY in the PWM Channel Duty Cycle Register (PWM_CDTYx) to generate an output signal OCx accordingly.

The different properties of the waveform of the output OCx are:

- the clock selection. The channel counter is clocked by one of the clocks provided by the clock generator described in the previous section. This channel parameter is defined in the CPRE field of the PWM Channel Mode Register (PWM_CMRx). This field is reset at ‘0’.
- the waveform period. This channel parameter is defined in the CPRD field of the PWM_CPRDx register.
  If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

  By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

  $$\frac{(X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

  If the waveform is center-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(2 \times X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

  By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

  $$\frac{(2 \times X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or}$$

  $$\frac{(2 \times X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$
- the waveform duty-cycle. This channel parameter is defined in the CDTY field of the PWM_CDTYx register.
  If the waveform is left-aligned, then:

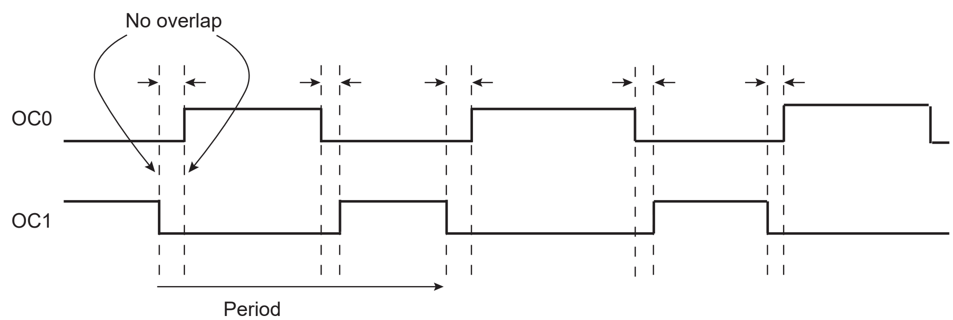
  $$\text{duty cycle} = \frac{\mathrm{PWM\_period} - ((1/\mathrm{clkN}) \times \mathrm{CDTY})}{\mathrm{PWM\_period}}$$

If the waveform is center-aligned, then:

$$\text{duty cycle} = \frac{((\text{PWM\_period}/2) - 1) - ((1/\text{clkN}) \times \text{CDTY})}{\text{PWM\_period}/2}$$

- the waveform polarity. At the beginning of the period, the signal can be at high or low level. This property is defined in the CPOL bit of PWM_CMRx. By default, the signal starts by a low level. The DPOLI bit in PWM_CMRx defines the PWM polarity when the channel is disabled (CHIDx = 0 in PWM_SR). For more details, see the figure *Waveform Properties.*
  - DPOLI = 0: PWM polarity when the channel is disabled is the same as the one defined for the beginning of the PWM period.
  - DPOLI = 1: PWM polarity when the channel is disabled is inverted compared to the one defined for the beginning of the PWM period.
- the waveform alignment. The output waveform can be left- or center-aligned. Center-aligned waveforms can be used to generate non-overlapped waveforms. This property is defined in the CALG bit of PWM_CMRx. The default mode is left-aligned.

Figure 68-4. Non-Overlapped Center-Aligned Waveforms



Note: See the figure Figure 68-5 for a detailed description of center-aligned waveforms.

When center-aligned, the channel counter increases up to CPRD and decreases down to 0. This ends the period.

When left-aligned, the channel counter increases up to CPRD and is reset. This ends the period.

Thus, for the same CPRD value, the period for a center-aligned channel is twice the period for a left-aligned channel.

Waveforms are fixed at 0 when:

- CDTY = CPRD and CPOL = 0 (Note that if TRGMODE = MODE3, the PWM waveform switches to 1 at the external trigger event (see Cycle-By-Cycle Duty Mode)).
- CDTY = 0 and CPOL = 1

Waveforms are fixed at 1 (once the channel is enabled) when:

- CDTY = 0 and CPOL = 0
- CDTY = CPRD and CPOL = 1 (Note that if TRGMODE = MODE3, the PWM waveform switches to 0 at the external trigger event (see Cycle-By-Cycle Duty Mode)).

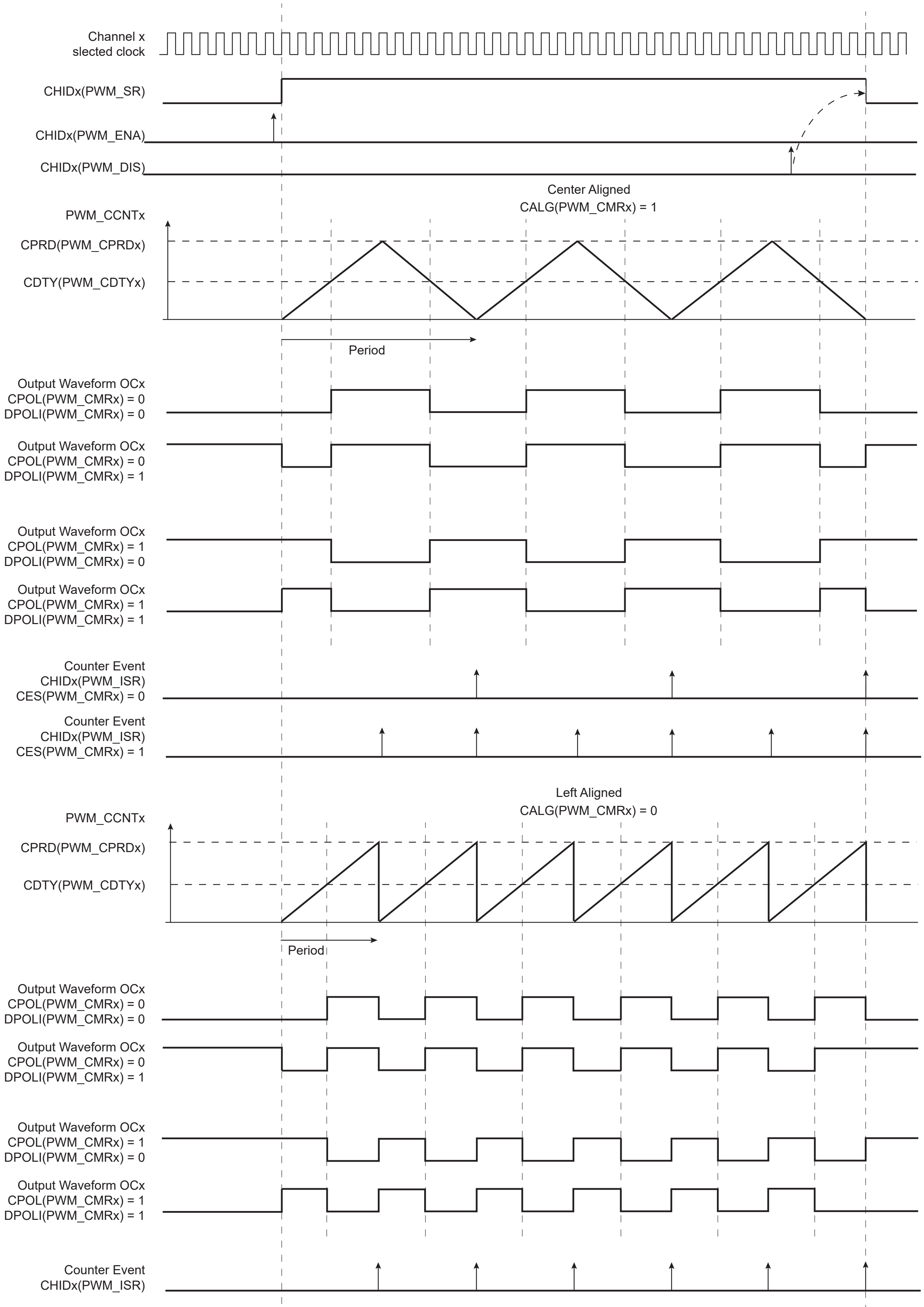
The waveform polarity must be set before enabling the channel. This immediately affects the channel output level.

Modifying CPOL in PWM Channel Mode Register while the channel is enabled can lead to an unexpected behavior of the device being driven by PWM.

In addition to generating the output signals OCx, the comparator generates interrupts depending on the counter value. When the output waveform is left-aligned, the interrupt occurs at the end of the counter period. When the output waveform is center-aligned, the bit CES of PWM_CMRx defines when the channel counter interrupt occurs. If CES is set to ‘0’, the interrupt occurs at the end of the counter period. If CES is set to ‘1’, the interrupt occurs at the end of the counter period and at half of the counter period.

The figure below illustrates the counter interrupts depending on the configuration.

Figure 68-5. Waveform Properties



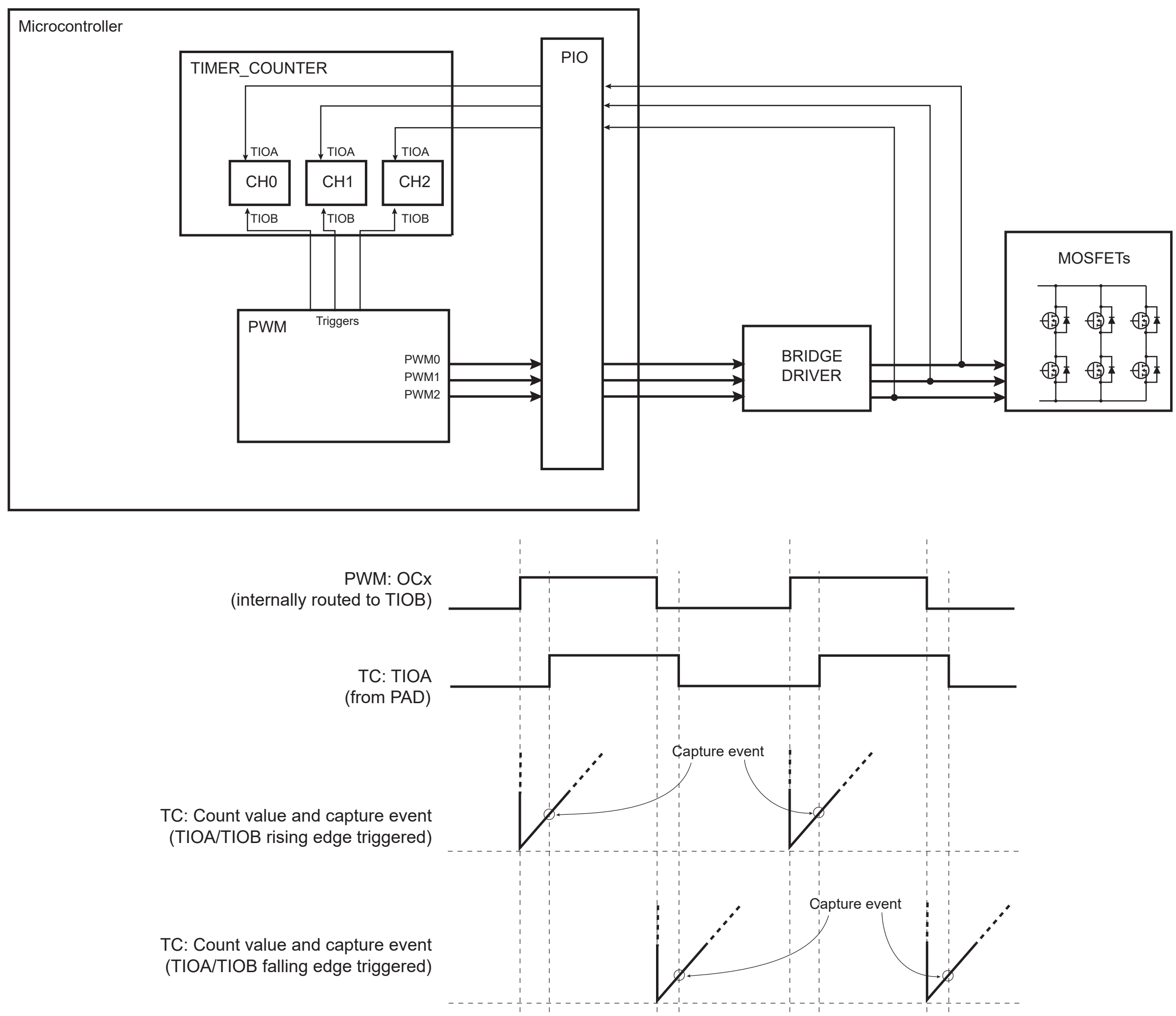
68.6.2.3 Trigger Selection for Timer Counter

The PWM controller can be used as a trigger source for the Timer Counter (TC) to achieve the two application examples described below.

68.6.2.3.1 Delay Measurement

To measure the delay between the channel x comparator output (OCx) and the feedback from the bridge driver of the MOSFETs (see the figure below), the bit TCTS in the PWM Channel Mode Register must be at 0. This defines the comparator output of the channel x as the TC trigger source. The TIOB trigger (TC internal input) is used to start the TC; the TIOA input (from PAD) is used to capture the delay.

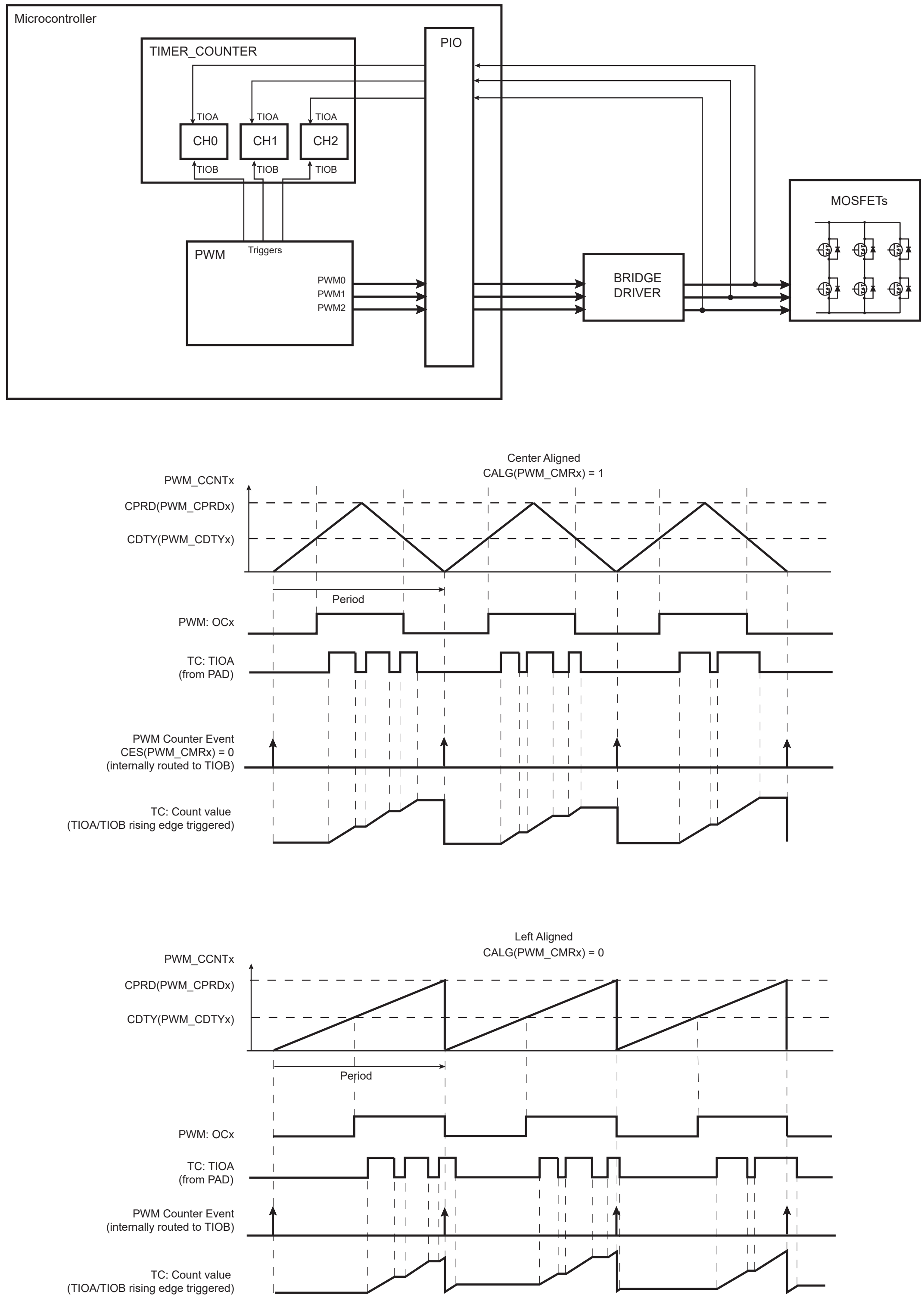
Figure 68-6. Triggering the TC: Delay Measurement



68.6.2.3.2 Cumulated ON Time Measurement

To measure the cumulated "ON" time of MOSFETs (see the figure below), the bit TCTS of the PWM Channel Mode Register must be set to 1 to define the counter event (see the figure *Waveform Properties*) as the Timer Counter trigger source.

Figure 68-7. Triggering the TC: Cumulated “ON” Time Measurement



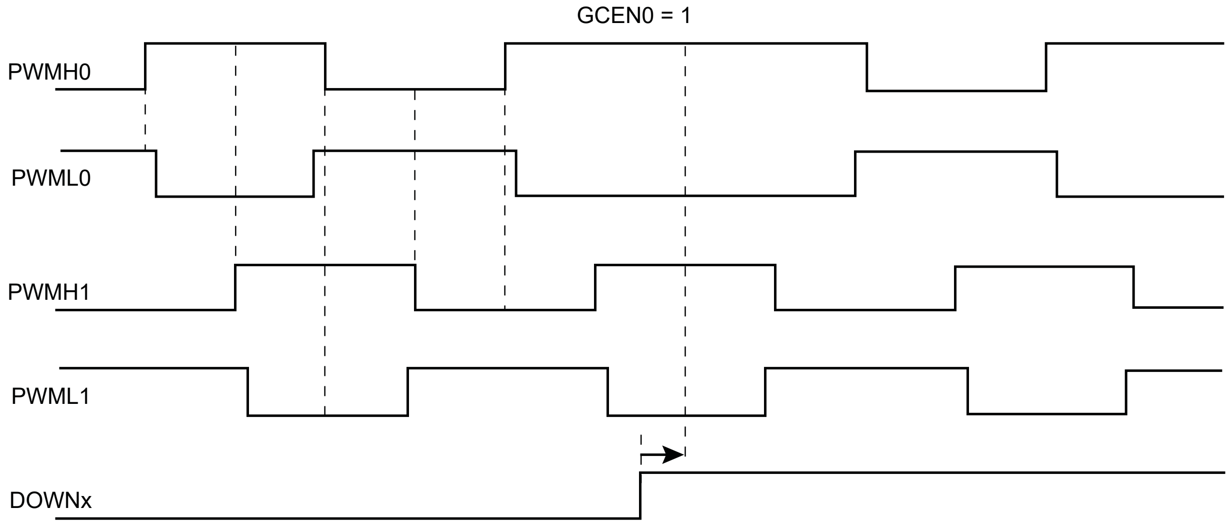
68.6.2.4 2-bit Gray Up/Down Counter for Stepper Motor

A pair of channels may provide a 2-bit Gray count waveform on two outputs. Dead-time generator and other downstream logic can be configured on these channels.

Up or Down Count mode can be configured on-the-fly by means of PWM_SMMR configuration registers.

When GCEN0 is set to ‘1’, channels 0 and 1 outputs are driven with a Gray counter.

Figure 68-8. 2-bit Gray Up/Down Counter



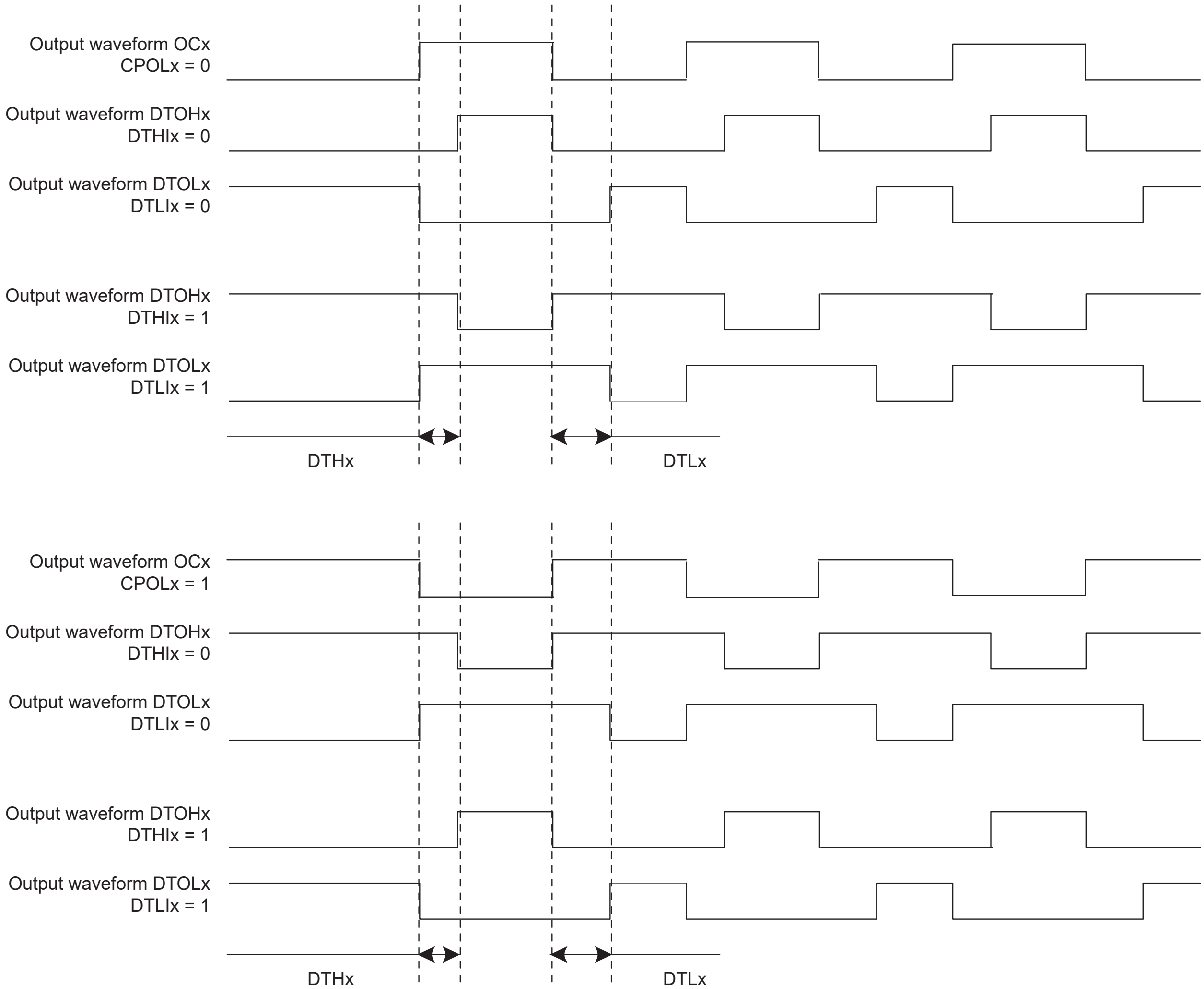
68.6.2.5 Dead-Time Generator

The dead-time generator uses the comparator output OCx to provide the two complementary outputs DTOHx and DTOLx, which allows the PWM macrocell to drive external power control switches safely. When the dead-time generator is enabled by setting the bit DTE to 1 or 0 in the PWM Channel Mode Register (PWM_CMRx), dead-times (also called dead-bands or non-overlapping times) are inserted between the edges of the two complementary outputs DTOHx and DTOLx. Note that enabling or disabling the dead-time generator is allowed only if the channel is disabled.

The dead-time is adjustable by the PWM Channel Dead Time Register (PWM_DTx). Each output of the dead-time generator can be adjusted separately by DTH and DTL. The dead-time values can be updated synchronously to the PWM period by using the PWM Channel Dead Time Update Register (PWM_DTUPDx).

The dead-time is based on a specific counter which uses the same selected clock that feeds the channel counter of the comparator. Depending on the edge and the configuration of the dead-time, DTOHx and DTOLx are delayed until the counter has reached the value defined by DTH or DTL. An inverted configuration bit (DTHI and DTLI bit in PWM_CMRx) is provided for each output to invert the dead-time outputs. The following figure shows the waveform of the dead-time generator.

Figure 68-9. Complementary Output Waveforms



68.6.2.5.1 PWM Push-Pull Mode

When a PWM channel is configured in Push-Pull mode, the dead-time generator output is managed alternately on each PWM cycle. The polarity of the PWM line during the idle state of the Push-Pull mode is defined by the DPOLI bit in the PWM Channel Mode Register (PWM_CMRx). The Push-Pull mode can be enabled separately on each channel by writing a one to bit PPM in the PWM Channel Mode Register.

Figure 68-10. PWM Push-Pull Mode

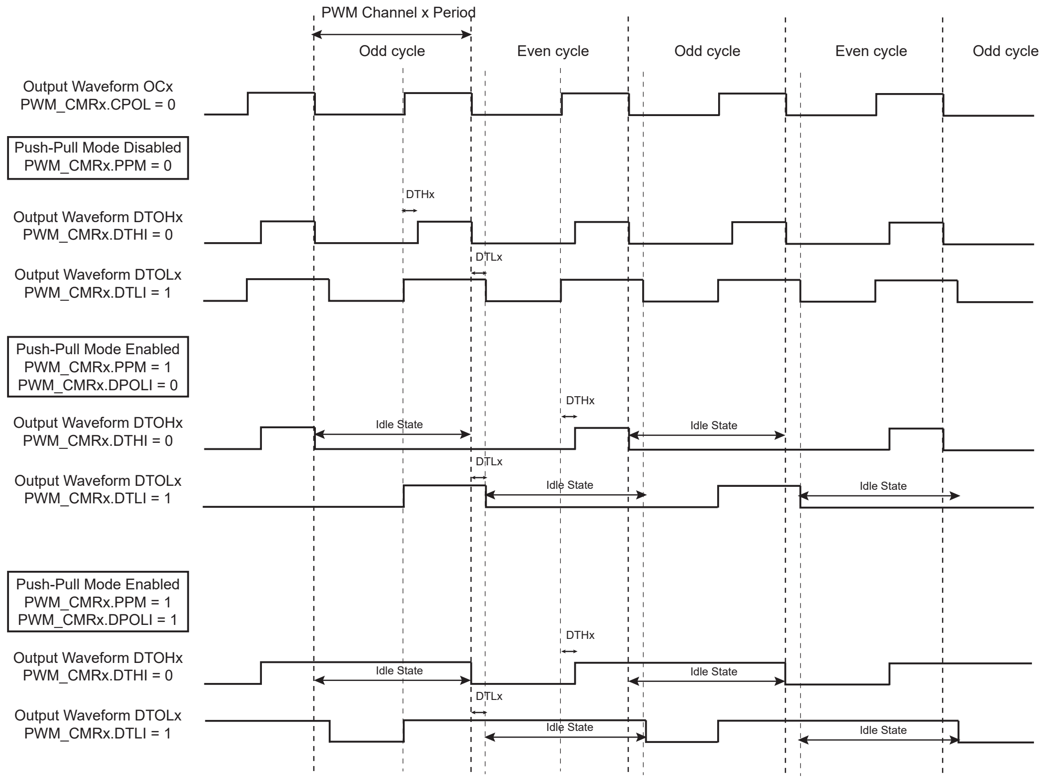


Figure 68-11. PWM Push-Pull Waveforms: Left-Aligned Mode

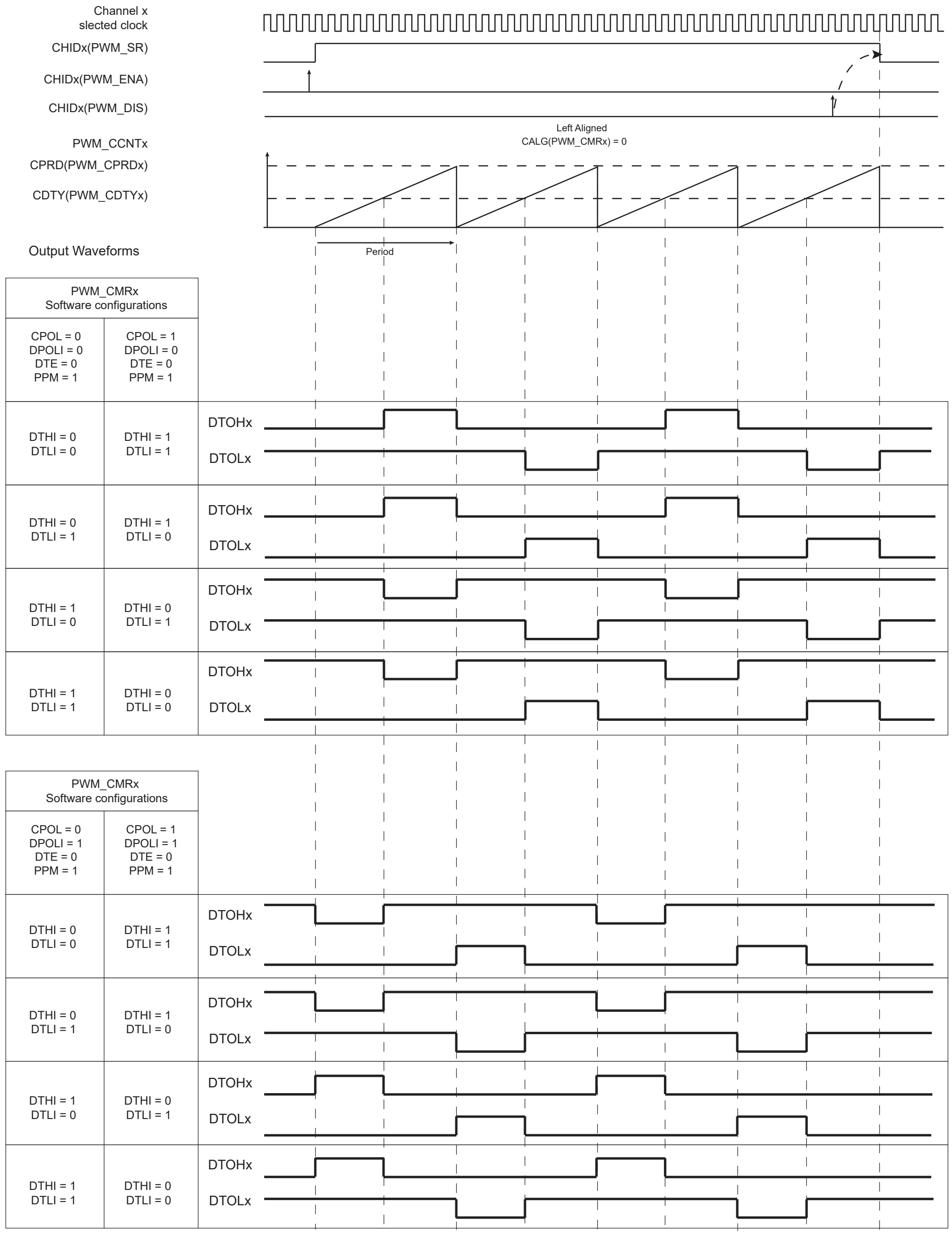
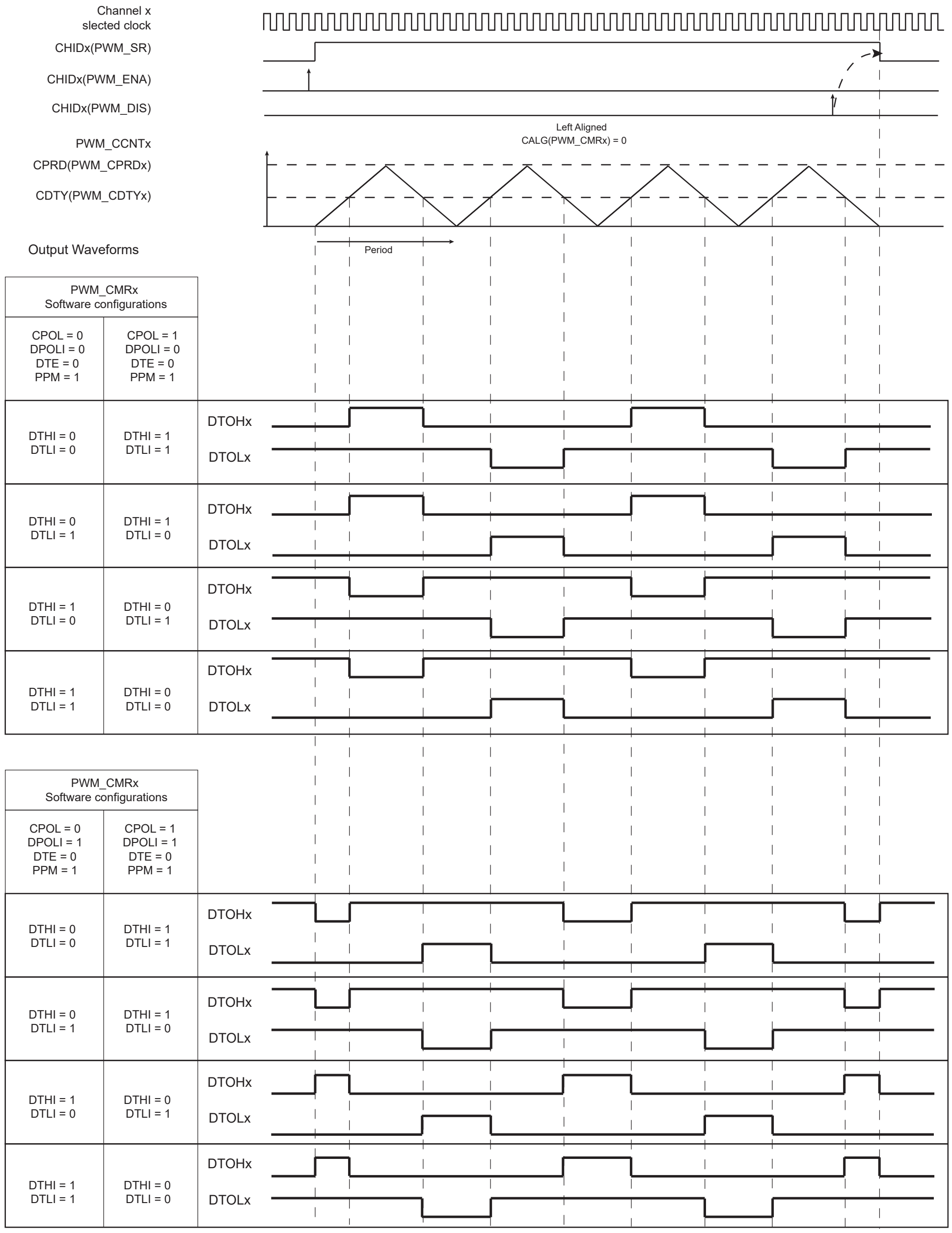


Figure 68-12. PWM Push-Pull Waveforms: Center-Aligned Mode



The PWM Push-Pull mode can be useful in transformer-based power converters, such as a half-bridge converter. The Push-Pull mode prevents the transformer core from being saturated by any direct current.

Figure 68-13. Half-Bridge Converter Application: No Feedback Regulation

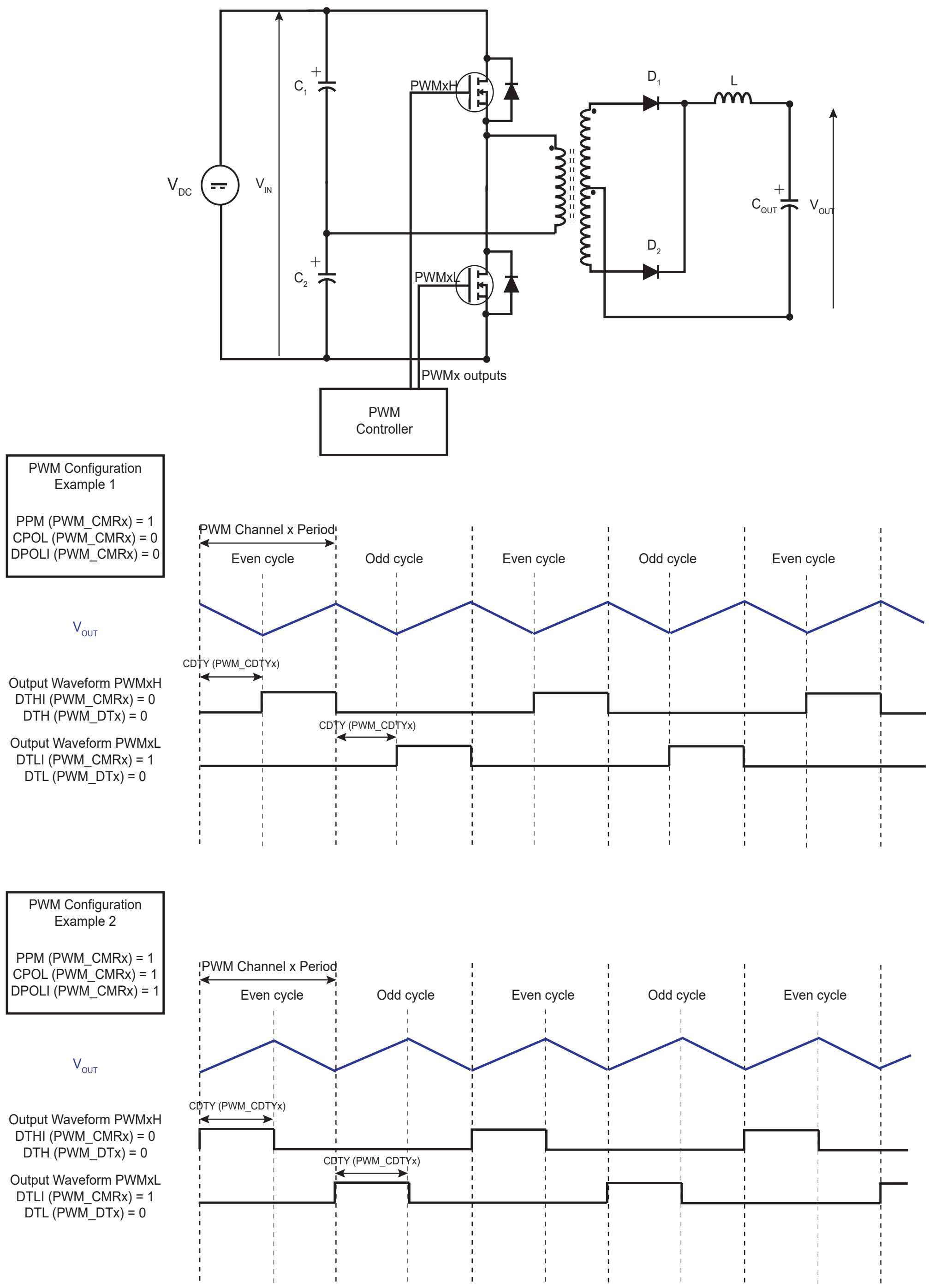
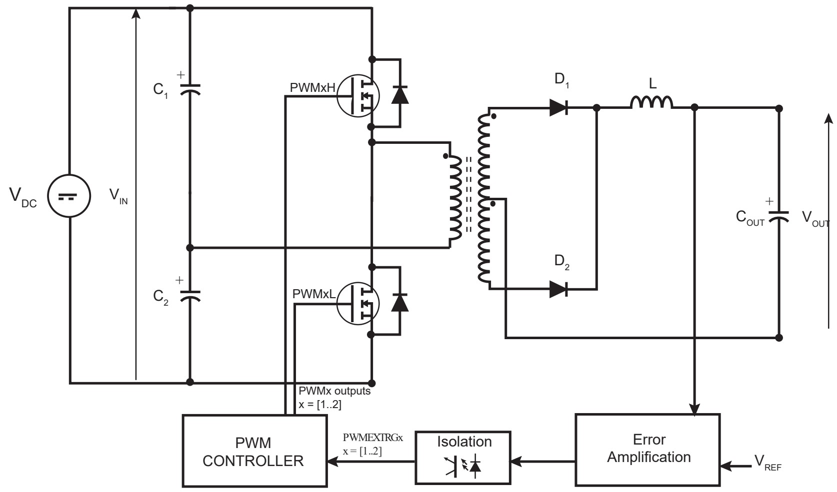
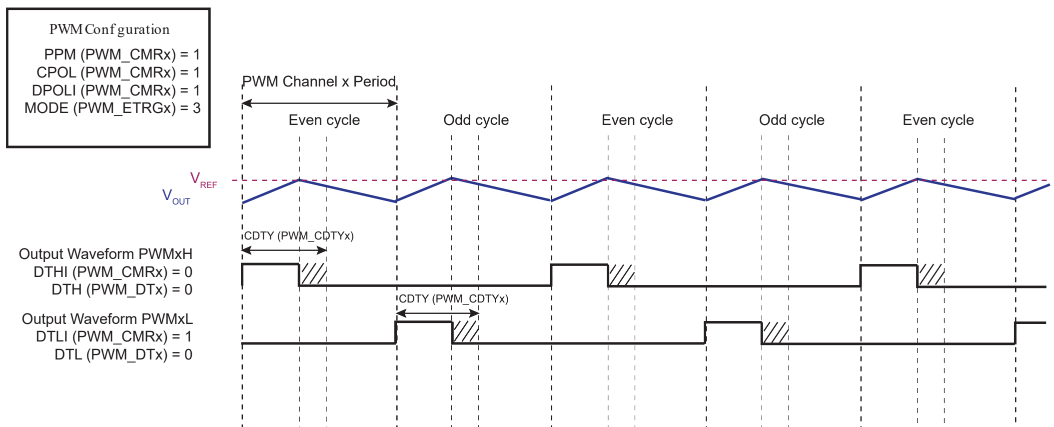


Figure 68-14. Half-Bridge Converter Application: Feedback Regulation

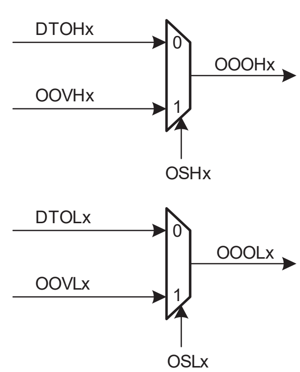




68.6.2.6 Output Override

The two complementary outputs DTOHx and DTOLx of the dead-time generator can be forced to a value defined by the software.

Figure 68-15. Override Output Selection



The fields OSHx and OSLx in the PWM Output Selection Register (PWM_OS) allow the outputs of the dead-time generator DTOHx and DTOLx to be overridden by the value defined in the fields OOVHx and OOVLx in the PWM Output Override Value Register (PWM_OOV).

The set registers PWM Output Selection Set Register (PWM_OSS) and PWM Output Selection Set Update Register (PWM_OSSUPD) enable the override of the outputs of a channel regardless of other channels. In the same way, the clear registers PWM Output Selection Clear Register (PWM_OSC) and PWM Output Selection Clear Update Register (PWM_OSCUPD) disable the override of the outputs of a channel regardless of other channels.

By using buffer registers PWM_OSSUPD and PWM_OSCUPD, the output selection of PWM outputs is done synchronously to the channel counter, at the beginning of the next PWM period.

By using registers PWM_OSS and PWM_OSC, the output selection of PWM outputs is done asynchronously to the channel counter, as soon as the register is written.

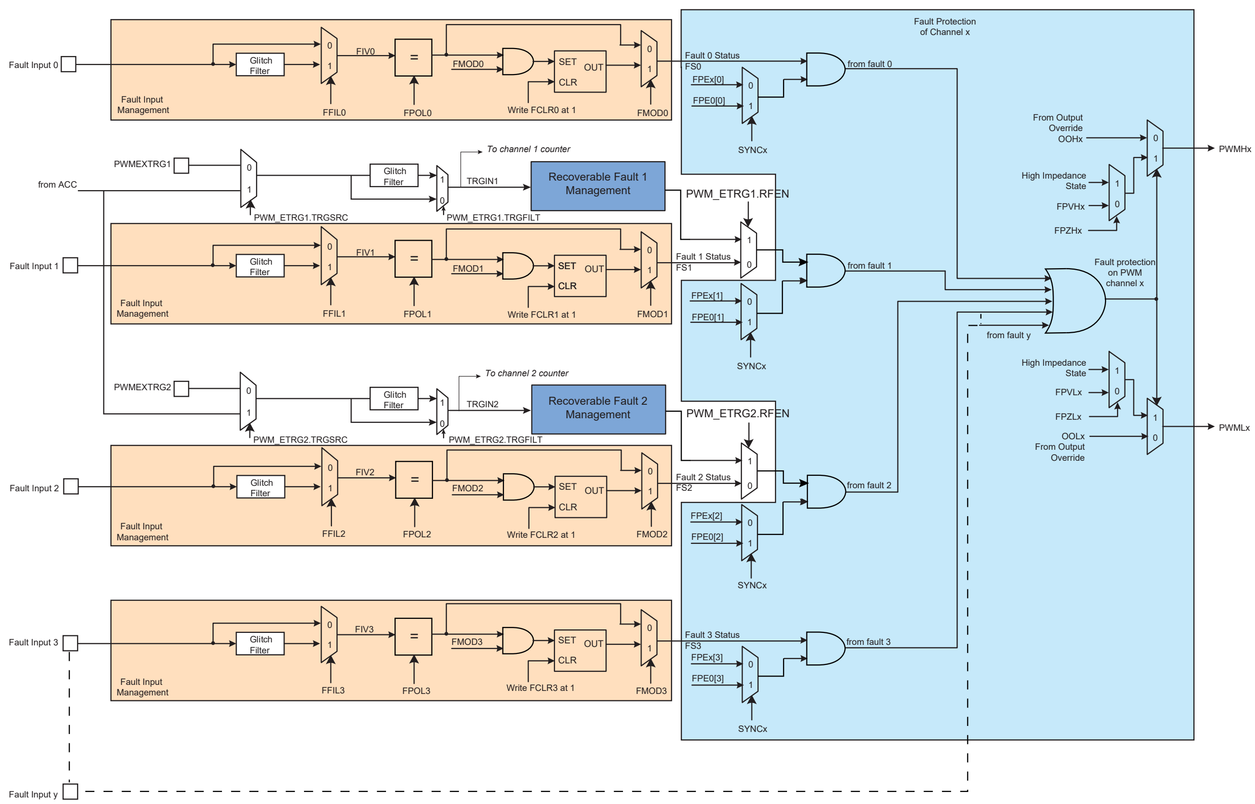
The value of the current output selection can be read in PWM_OS.

While overriding PWM outputs, the channel counters continue to run, only the PWM outputs are forced to user defined values.

68.6.2.7 Fault Protection

6 inputs provide fault protection which can force any of the PWM output pairs to a programmable value. This mechanism has priority over output overriding.

Figure 68-16. Fault Protection



The polarity level of the fault inputs is configured by the FPOL field in the PWM Fault Mode Register (PWM_FMR). For fault inputs coming from internal peripherals such as ADC or Timer Counter, the polarity level must be FPOL = 1. For fault inputs coming from external GPIO pins the polarity level depends on the user's implementation.

The configuration of the Fault Activation mode (FMOD field in PWMC_FMR) depends on the peripheral generating the fault. If the corresponding peripheral does not have “Fault Clear” management, then the FMOD configuration to use must be FMOD = 1, to avoid spurious fault detection. Refer to the corresponding peripheral documentation for details on handling fault generation.

Fault inputs may or may not be glitch-filtered depending on the FFIL field in PWM_FMR. When the filter is activated, glitches on fault inputs with a width inferior to the PWM peripheral clock period are rejected.

A fault becomes active as soon as its corresponding fault input has a transition to the programmed polarity level. If the corresponding bit FMOD is set to ‘0’ in PWM_FMR, the fault remains active as long as the fault input is at this polarity level. If the corresponding FMOD field is set to ‘1’, the fault remains active until the fault input is no longer at this polarity level and until it is cleared by writing the corresponding bit FCLR in the PWM Fault Clear Register

(PWM_FCR). In the PWM Fault Status Register (PWM_FSR), the field FIV indicates the current level of the fault inputs and the field FIS indicates whether a fault is currently active.

Each fault can be taken into account or not by the fault protection mechanism in each channel. To be taken into account in the channel x, the fault y must be enabled by the bit FPEx[y] in the PWM Fault Protection Enable register (PWM_FPE). However, synchronous channels (see Synchronous Channels) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[y]).

The fault protection on a channel is triggered when this channel is enabled and when any one of the faults that are enabled for this channel is active. It can be triggered even if the PWM peripheral clock is not running but only by a fault input that is not glitch-filtered.

When the fault protection is triggered on a channel, the fault protection mechanism resets the counter of this channel and forces the channel outputs to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV) and fields FPZHx/FPZLx in the PWM Fault Protection Value Register 2, as shown in the table below. The output forcing is made asynchronously to the channel counter.

Table 68-4. Forcing Values of PWM Outputs by Fault Protection

FPZH/Lx	FPVH/Lx	Forcing Value of PWMH/Lx
0	0	0
0	1	1
1	–	High impedance state (Hi-Z)

⚠ CAUTION

- To prevent any unexpected activation of the status flag FSy in PWM_FSR, the FMODy bit can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.
- To prevent any unexpected activation of the Fault Protection on the channel x, the bit FPEx[y] can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

If a comparison unit is enabled (see PWM Comparison Units) and if a fault is triggered in the channel 0, then the comparison cannot match.

As soon as the fault protection is triggered on a channel, an interrupt (different from the interrupt generated at the end of the PWM period) can be generated but only if it is enabled and not masked. The interrupt is reset by reading the interrupt status register, even if the fault which has caused the trigger of the fault protection is kept active.

68.6.2.7.1 Recoverable Fault

The PWM provides a Recoverable Fault mode on fault 1 and 2 (see figure Fault Protection).

The recoverable fault signal is an internal signal generated as soon as an external trigger event occurs (see PWM External Trigger Mode).

When the fault 1 or 2 is defined as a recoverable fault, the corresponding fault input pin is ignored and bits FFIL1/2, FMOD1/2 and FFIL1/2 are not taken into account.

The fault 1 is managed as a recoverable fault by the PWMEXTRG1 input trigger when PWM_ETRG1.RFEN = 1, PWM_ENA.CHID1 = 1, and PWM_ETRG1.TRGMODE ≠ 0.

The fault 2 is managed as a recoverable fault by the PWMEXTRG2 input trigger when PWM_ETRG2.RFEN = 1, PWM_ENA.CHID2 = 1, and PWM_ETRG2.TRGMODE ≠ 0.

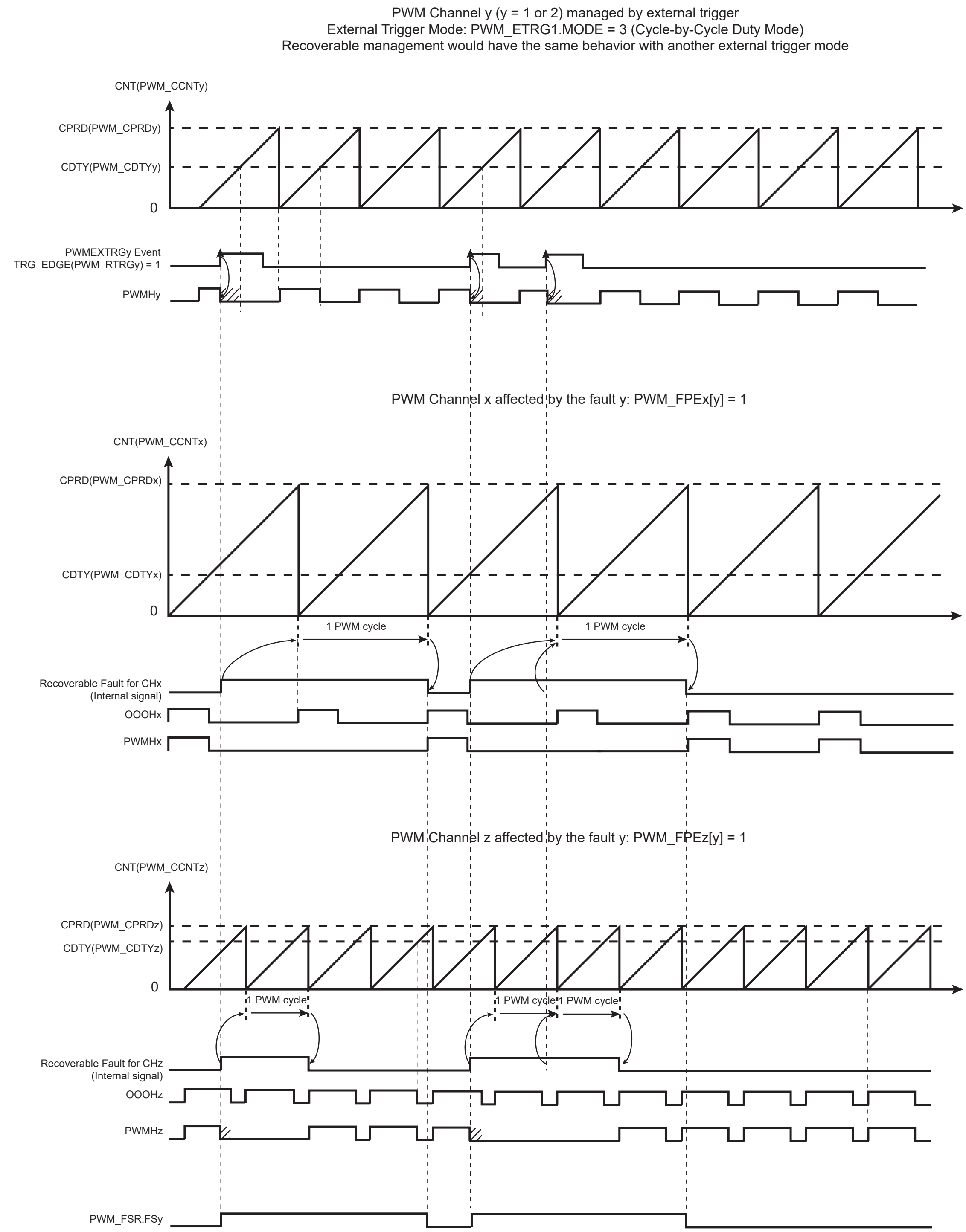
Recoverable fault 1 and 2 can be taken into account by all channels by enabling the bit FPEx[1/2] in the PWM Fault Protection Enable registers (PWM_FPEx). However the synchronous channels (see Synchronous Channels) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[1/2]).

When a recoverable fault is triggered (according to the PWM_ETRGx.TRGMODE setting), the PWM counter of the affected channels is not cleared (unlike in the classic fault protection mechanism) but the channel outputs are forced to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV), as per table *Forcing Values of PWM Outputs by Fault Protection*. The output forcing is made asynchronously to the channel counter and lasts from the recoverable fault occurrence to the end of the next PWM cycle (if the recoverable fault is no longer present) (see the figure below).

The recoverable fault does not trigger an interrupt. The Fault Status FSy (with y = 1 or 2) is not reported in the PWM Fault Status Register when the fault y is a recoverable fault.

See Cycle-By-Cycle Duty Mode: LED String Control for an application case associating the Recoverable Fault mode with the External Trigger mode.

Figure 68-17. Recoverable Fault Management



68.6.2.8 Spread Spectrum Counter

The PWM macrocell includes a spread spectrum counter allowing the generation of a constantly varying duty cycle on the output PWM waveform (only for the channel 0). This feature may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

This is achieved by varying the effective period in a range defined by a spread spectrum value which is programmed by the field SPRD in the PWM Spread Spectrum Register (PWM_SSPR). The effective period of the output waveform is the value of the spread spectrum counter added to the programmed waveform period CPRD in the PWM Channel Period Register (PWM_CPRD0).

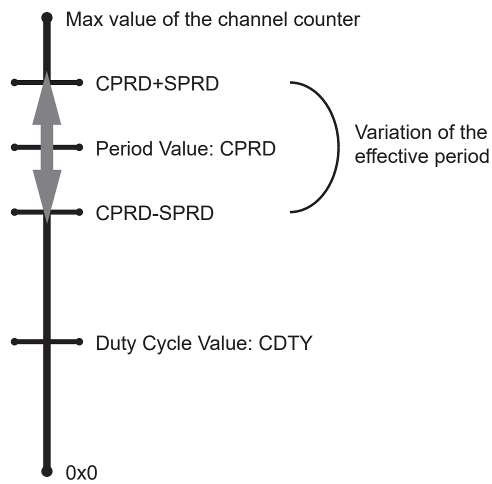
It will cause the effective period to vary from CPRD-SPRD to CPRD+SPRD. This leads to a constantly varying duty cycle on the PWM output waveform because the duty cycle value programmed is unchanged.

The value of the spread spectrum counter can change in two ways depending on the bit SPRDM in PWM_SSPR.

If SPRDM = 0, the Triangular mode is selected. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled or after reset and counts upwards at each period of the channel counter. When it reaches SPRD, it restarts to count from -SPRD again.

If SPRDM = 1, the Random mode is selected. A new random value is assigned to the spread spectrum counter at each period of the channel counter. This random value is between -SPRD and +SPRD and is uniformly distributed.

Figure 68-18. Spread Spectrum Counter



68.6.2.9 Synchronous Channels

Some channels can be linked together as synchronous channels. They have the same source clock, the same period, the same alignment and are started together. In this way, their counters are synchronized together.

The synchronous channels are defined by the SYNCx bits in the PWM Sync Channels Mode Register (PWM_SCM). Only one group of synchronous channels is allowed.

When a channel is defined as a synchronous channel, the channel 0 is also automatically defined as a synchronous channel. This is because the channel 0 counter configuration is used by all the synchronous channels.

If a channel x is defined as a synchronous channel, the fields/bits for the channel 0 are used instead of those of channel x:

- CPRE in PWM_CMR0 instead of CPRE in PWM_CMRx (same source clock)
- CPRD in PWM_CPRD0 instead of CPRD in PWM_CPRDx (same period)
- CALG in PWM_CMR0 instead of CALG in PWM_CMRx (same alignment)

Modifying the fields CPRE, CPRD and CALG of for channels with index greater than 0 has no effect on output waveforms.

Because counters of synchronous channels must start at the same time, they are all enabled together by enabling the channel 0 (by the CHID0 bit in PWM_ENA register). In the same way, they are all disabled together by disabling channel 0 (by the CHID0 bit in PWM_DIS register). However, a synchronous channel x different from channel 0 can be enabled or disabled independently from others (by the CHIDx bit in PWM_ENA and PWM_DIS registers).

Defining a channel as a synchronous channel while it is an asynchronous channel (by writing the bit SYNCx to ‘1’ while it was at ‘0’) is allowed only if the channel is disabled at this time (CHIDx = 0 in PWM_SR). In the same way, defining a channel as an asynchronous channel while it is a synchronous channel (by writing the SYNCx bit to ‘0’ while it was ‘1’) is allowed only if the channel is disabled at this time.

The UPDM field (Update Mode) in the PWM_SCM register selects one of the three methods to update the registers of the synchronous channels:

- Method 1 (UPDM = 0): The period value, the duty-cycle values and the dead-time values must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).The update is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM Sync Channels Update Control Register (PWM_SCUC) is set to ‘1’.
- Method 2 (UPDM = 1): The period value, the duty-cycle values, the dead-time values and the update period value must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPD). The update of the period value and of the dead-time values is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM_SCUC register is set to ‘1’. The update of the duty-cycle values and the update period value is triggered automatically after an update period defined by the field UPR in the PWM Sync Channels Update Period Register (PWM_SCUP).
- Method 3 (UPDM = 2): Same as Method 2 apart from the fact that the duty-cycle values of ALL synchronous channels are written by the DMA Controller. The user can choose to synchronize the DMA Controller transfer request with a comparison match (see Section 7.3 “PWM Comparison Units”), by the fields PTRM and PTRCS in the PWM_SCM register. The DMA destination address must be configured to access only the PWM DMA Register (PWM_DMAR). The DMA buffer data structure must consist of sequentially repeated duty cycles. The number of duty cycles in each sequence corresponds to the number of synchronized channels. Duty cycles in each sequence must be ordered from the lowest to the highest channel index. The size of the duty cycle is 16 bits.

Table 68-5. Summary of the Update of Registers of Synchronous Channels

<table>
<tr><th>Register</th><th>UPDM = 0</th><th>UPDM = 1</th><th>UPDM = 2</th></tr>
<tr><td rowspan="2">Period Value (PWM_CPRDUPDx)</td><td colspan="3">Write by the processor</td></tr>
<tr><td colspan="3">Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to ‘1’</td></tr>
<tr><td rowspan="2">Dead-Time Values (PWM_DTUPDx)</td><td colspan="3">Write by the processor</td></tr>
<tr><td colspan="3">Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to ‘1’</td></tr>
<tr><td rowspan="2">Duty-Cycle Values (PWM_CDTYUPDx)</td><td>Write by the processor</td><td>Write by the processor</td><td>Write by the DMA Controller</td></tr>
<tr><td>Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to ‘1’</td><td colspan="2">Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR</td></tr>
<tr><td rowspan="2">Update Period Value (PWM_SCUPUPD)</td><td>Not applicable</td><td colspan="2">Write by the processor</td></tr>
<tr><td>Not applicable</td><td colspan="2">Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR</td></tr>
</table>

68.6.2.9.1 Method 1: Manual write of duty-cycle values and manual trigger of the update

In this mode, the update of the period value, the duty-cycle values and the dead-time values must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).

To trigger the update, the user must use the bit UPDULOCK in the PWM_SCUC register which allows to update synchronously (at the same PWM period) the synchronous channels:

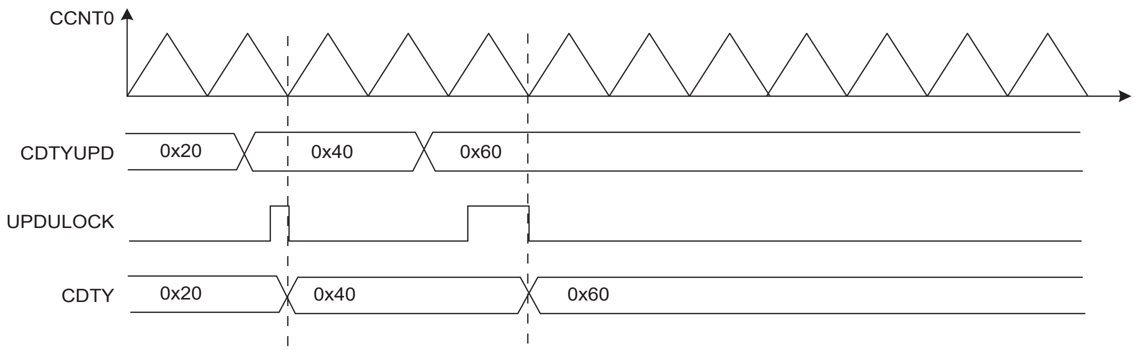
- If the bit UPDULOCK is set to ‘1’, the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to ‘1’, the update is locked and cannot be performed.

After writing the UPDULOCK bit to ‘1’, it is held at this value until the update occurs, then it is read 0.

Sequence for Method 1:

1. Select the manual write of duty-cycle values and the manual update by setting the UPDM field to ‘0’ in the PWM_SCM register.
2. Define the synchronous channels by the SYNCx bits in the PWM_SCM register.
3. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
4. If an update of the period value and/or the duty-cycle values and/or the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).
5. Set UPDULOCK to ‘1’ in PWM_SCUC.
6. The update of the registers will occur at the beginning of the next PWM period. When the UPDULOCK bit is reset, go to Step 4. for new values.

Figure 68-19. Method 1 (UPDM = 0)



68.6.2.9.2 Method 2: Manual write of duty-cycle values and automatic trigger of the update

In this mode, the update of the period value, the duty-cycle values, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK in the PWM_SCUC register, which updates synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to ‘1’, the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to ‘1’, the update is locked and cannot be performed.

After writing the UPDULOCK bit to ‘1’, it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the update period by the UPR field in the PWM_SCUP register. The PWM controller waits UPR+1 period of synchronous channels before updating automatically the duty values and the update period value.

The status of the duty-cycle value write is reported in the PWM Interrupt Status Register 2 (PWM_ISR2) by the following flags:

- WRDY: this flag is set to ‘1’ when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to ‘0’ when the PWM_ISR2 register is read.

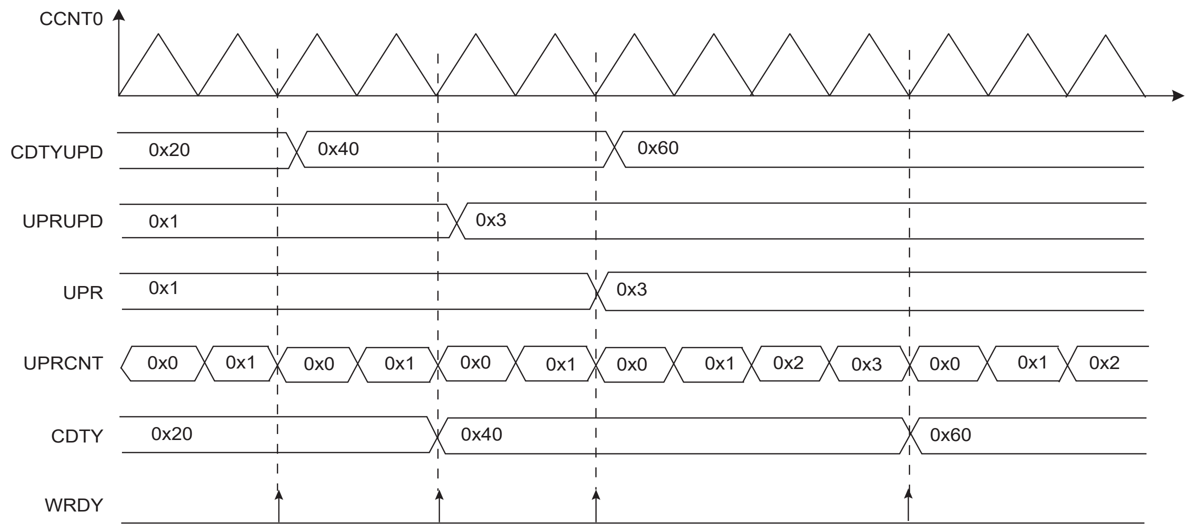
Depending on the interrupt mask in the PWM Interrupt Mask Register 2 (PWM_IMR2), an interrupt can be generated by these flags.

Sequence for Method 2:

1. Select the manual write of duty-cycle values and the automatic update by setting the field UPDM to ‘1’ in the PWM_SCM register
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
5. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 8.
6. Set UPDULOCK to ‘1’ in PWM_SCUC.

7. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 5. for new values.
8. If an update of the duty-cycle values and/or the update period is required, check first that write of new update values is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2.
9. Write registers that need to be updated (PWM_CDTYUPDx, PWM_SCUPUPD).
10. The update of these registers will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 8. for new values.

Figure 68-20. Method 2 (UPDM = 1)



68.6.2.9.3 Method 3: Automatic write of duty-cycle values and automatic trigger of the update

In this mode, the update of the duty cycle values is made automatically by the DMA Controller. The update of the period value, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK which allows to update synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period value is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the Update Period by the field UPR in the PWM_SCUP register. The PWM controller waits UPR+1 periods of synchronous channels before updating automatically the duty values and the update period value.

Using the DMA Controller removes processor overhead by reducing its intervention during the transfer. This significantly reduces the number of clock cycles required for a data transfer, which improves microcontroller performance.

The DMA Controller must write the duty-cycle values in the synchronous channels index order. For example if the channels 0, 1 and 3 are synchronous channels, the DMA Controller must write the duty-cycle of the channel 0 first, then the duty-cycle of the channel 1, and finally the duty-cycle of the channel 3.

The status of the DMA Controller transfer is reported in PWM_ISR2 by the following flags:

- WRDY: this flag is set to '1' when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to '0' when PWM_ISR2 is read. The user can choose to synchronize the WRDY flag and the DMA Controller transfer request with a comparison match (see PWM Comparison Units), by the fields PTRM and PTRCS in the PWM_SCM register.
- UNRE: this flag is set to '1' when the update period defined by the UPR field has elapsed while the whole data has not been written by the Peripheral DMA Controller. It is reset to '0' when PWM_ISR2 is read.

Depending on the interrupt mask in PWM_IMR2, an interrupt can be generated by these flags.

Sequence for Method 3:

1. Select the automatic write of duty-cycle values and automatic update by setting the field UPDM to 2 in the PWM_SCM register.
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Define when the WRDY flag and the corresponding DMA Controller transfer request must be set in the update period by the PTRM bit and the PTRCS field in the PWM_SCM register (at the end of the update period or when a comparison matches).
5. Define the DMA Controller transfer settings for the duty-cycle values and enable it in the DMA Controller registers
6. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
7. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 10.
8. Set UPDULOCK to '1' in PWM_SCUC.
9. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 7. for new values.
10. If an update of the update period value is required, check first that write of a new update value is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2, else go to Step 14.
11. Write the register that needs to be updated (PWM_SCUPUPD).
12. The update of this register will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 10 for new values.
13. Wait for the DMA status flag indicating that the buffer transfer is complete. If the transfer has ended, define a new DMA transfer for new duty-cycle values. Go to Step 5.

Figure 68-21. Method 3 (UPDM = 2 and PTRM = 0)

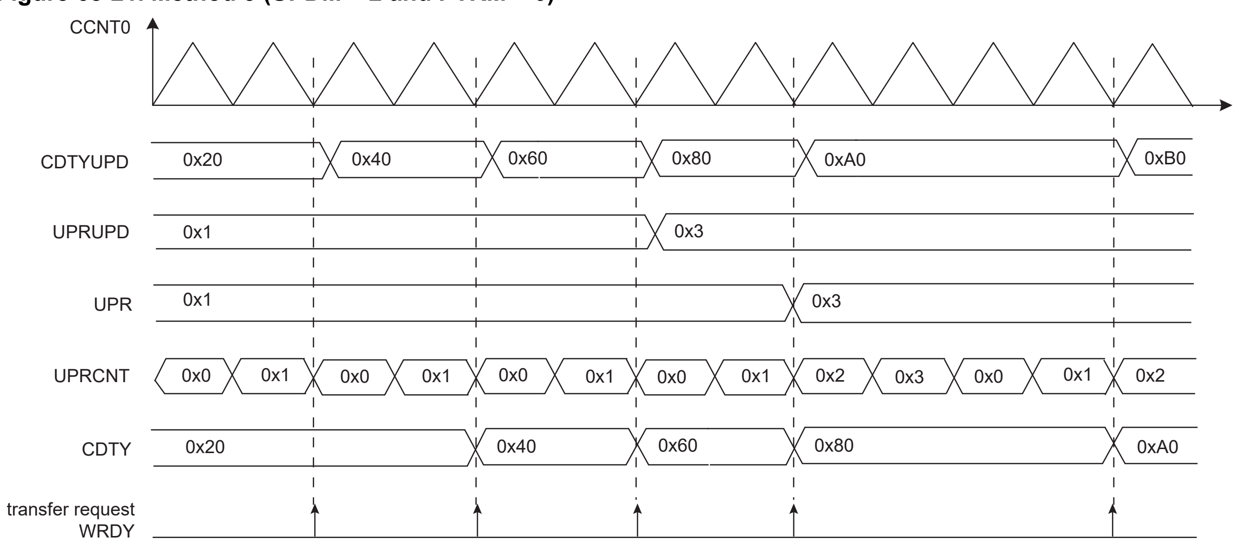
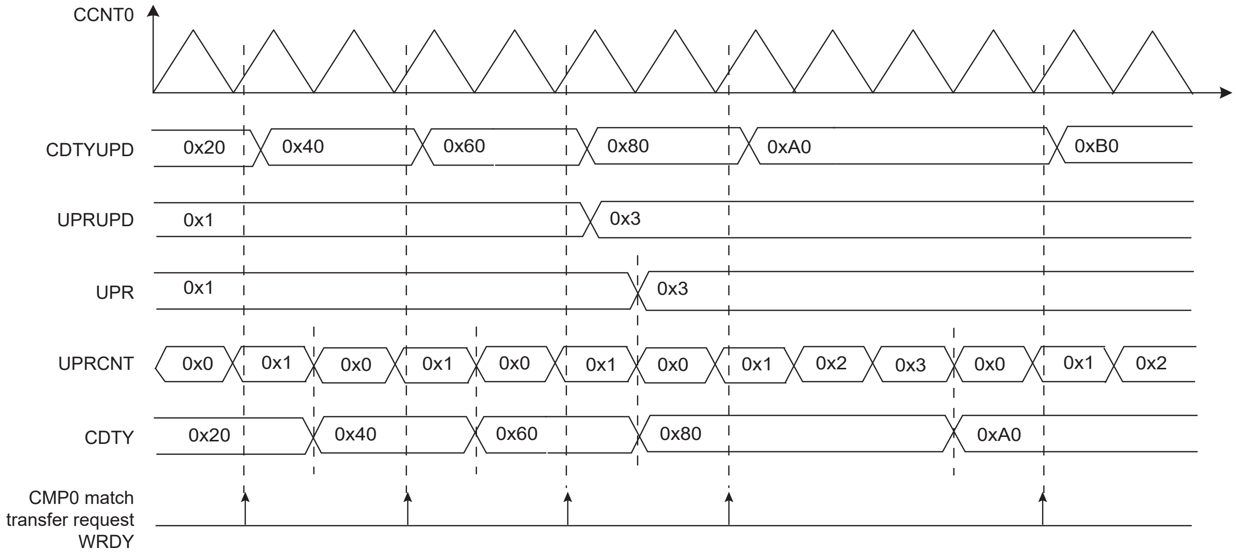


Figure 68-22. Method 3 (UPDM = 2 and PTRM = 1 and PTRCS = 0)



68.6.2.10 Update Time for Double-Buffering Registers

All channels integrate a double-buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum value, the polarity, the duty-cycle, the dead-times, the output override, and the synchronous channels update period.

This double-buffering system comprises the following update registers:

- PWM Sync Channels Update Period Update Register
- PWM Output Selection Set Update Register
- PWM Output Selection Clear Update Register
- PWM Spread Spectrum Update Register
- PWM Channel Duty Cycle Update Register
- PWM Channel Period Update Register
- PWM Channel Dead Time Update Register
- PWM Channel Mode Update Register

When one of these update registers is written to, the write is stored, but the values are updated only at the next PWM period border. In Left-aligned mode (CALG = 0), the update occurs when the channel counter reaches the period value CPRD. In Center-aligned mode, the update occurs when the channel counter value is decremented and reaches the 0 value.

In Center-aligned mode, it is possible to trigger the update of the polarity and the duty-cycle at the next half period border. This mode concerns the following update registers:

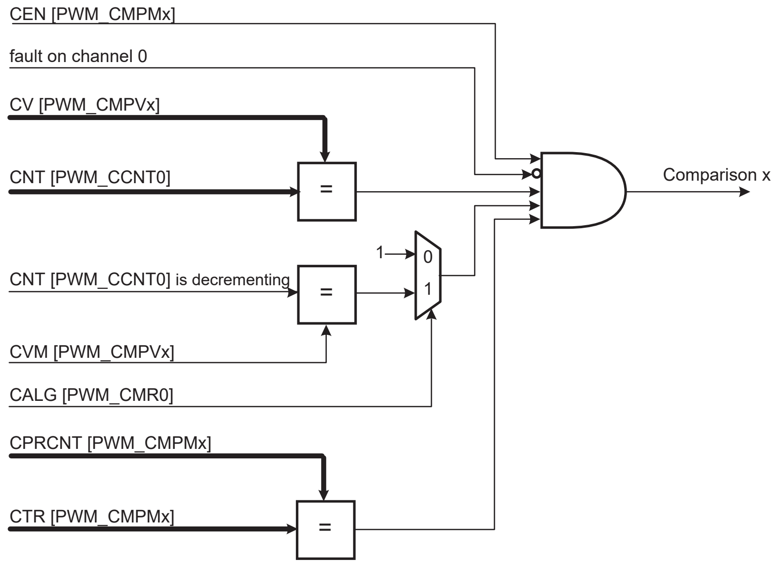
- PWM Channel Duty Cycle Update Register
- PWM Channel Mode Update Register

The update occurs at the first half period following the write of the update register (either when the channel counter value is incrementing and reaches the period value CPRD, or when the channel counter value is decrementing and reaches the 0 value). To activate this mode, the user must write a one to the bit UPDS in the PWM Channel Mode Register.

68.6.3 PWM Comparison Units

The PWM provides 8 independent comparison units able to compare a programmed value with the current value of the channel 0 counter (which is the channel counter of all synchronous channels, “Synchronous Channels”). These comparisons are intended to generate pulses on the event lines (used to synchronize ADC, see PWM Event Lines), to generate software interrupts and to trigger DMA Controller transfer requests for the synchronous channels (see Method 3: Automatic write of duty-cycle values and automatic trigger of the update).

Figure 68-23. Comparison Unit Block Diagram



The comparison x matches when it is enabled by the bit CEN in the PWM Comparison x Mode Register (PWM_CMPMx for the comparison x) and when the counter of the channel 0 reaches the comparison value defined by the field CV in PWM Comparison x Value Register (PWM_CMPVx for the comparison x). If the counter of the channel 0 is center-aligned (CALG = 1 in PWM Channel Mode Register), the bit CVM in PWM_CMPVx defines if the comparison is made when the counter is counting up or counting down (in Left-alignment mode CALG = 0, this bit is useless).

If a fault is active on the channel 0, the comparison is disabled and cannot match (see Fault Protection).

The user can define the periodicity of the comparison x by the fields CTR and CPR in PWM_CMPMx. The comparison is performed periodically once every CPR+1 periods of the counter of the channel 0, when the value of the comparison period counter CPRCNT in PWM_CMPMx reaches the value defined by CTR. CPR is the maximum value of the comparison period counter CPRCNT. If CPR = CTR = 0, the comparison is performed at each period of the counter of the channel 0.

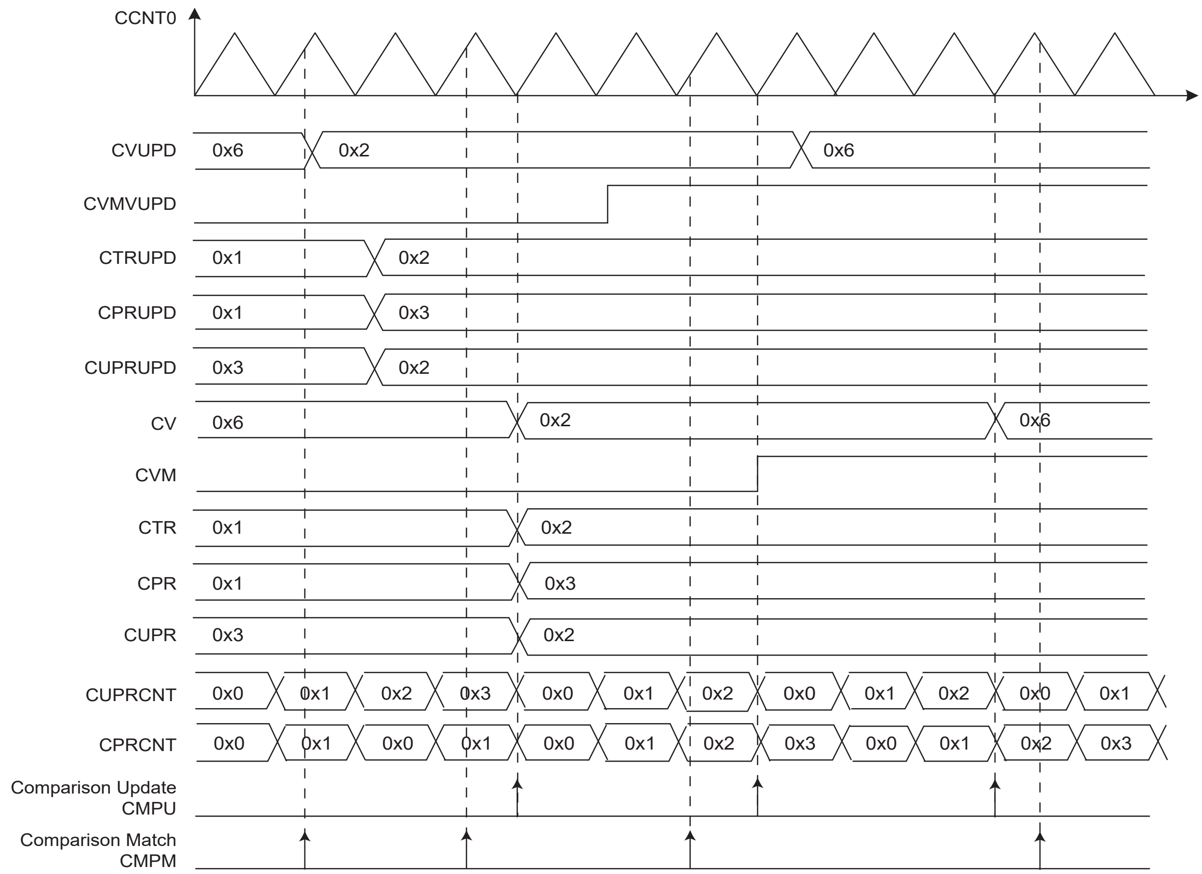
The comparison x configuration can be modified while the channel 0 is enabled by using the PWM Comparison x Mode Update Register (PWM_CMPMUPDx registers for the comparison x). In the same way, the comparison x value can be modified while the channel 0 is enabled by using the PWM Comparison x Value Update Register (PWM_CMPVUPDx registers for the comparison x).

The update of the comparison x configuration and the comparison x value is triggered periodically after the comparison x update period. It is defined by the field CUPR in PWM_CMPMx. The comparison unit has an update period counter independent from the period counter to trigger this update. When the value of the comparison update period counter CUPRCNT (in PWM_CMPMx) reaches the value defined by CUPR, the update is triggered. The comparison x update period CUPR itself can be updated while the channel 0 is enabled by using the PWM_CMPMUPDx register.

⚠ **CAUTION** The write of PWM_CMPVUPDx must be followed by a write of PWM_CMPMUPDx.

The comparison match and the comparison update can be source of an interrupt, but only if it is enabled and not masked. These interrupts can be enabled by the PWM Interrupt Enable Register 2 and disabled by the PWM Interrupt Disable Register 2. The comparison match interrupt and the comparison update interrupt are reset by reading the PWM Interrupt Status Register 2.

Figure 68-24. Comparison Waveform



68.6.4 PWM Event Lines

The PWM provides 2 independent event lines intended to trigger actions in other peripherals (e.g., for the Analog-to-Digital Converter (ADC)).

A pulse (one cycle of the peripheral clock) is generated on an event line, when at least one of the selected comparisons is matching. The comparisons can be selected or unselected independently by the CSEL bits in the PWM Event Line x Register (PWM_ELMRx for the Event Line x).

An example of event generation is provided in the figure Event Line Generation Waveform (Example).

Figure 68-25. Event Line Block Diagram

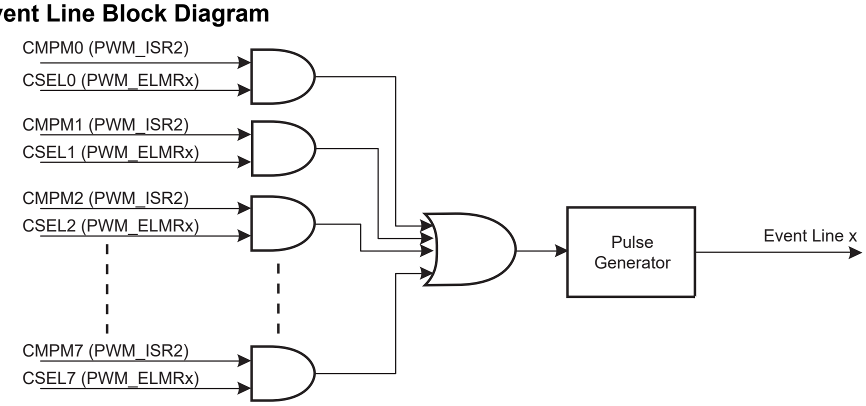
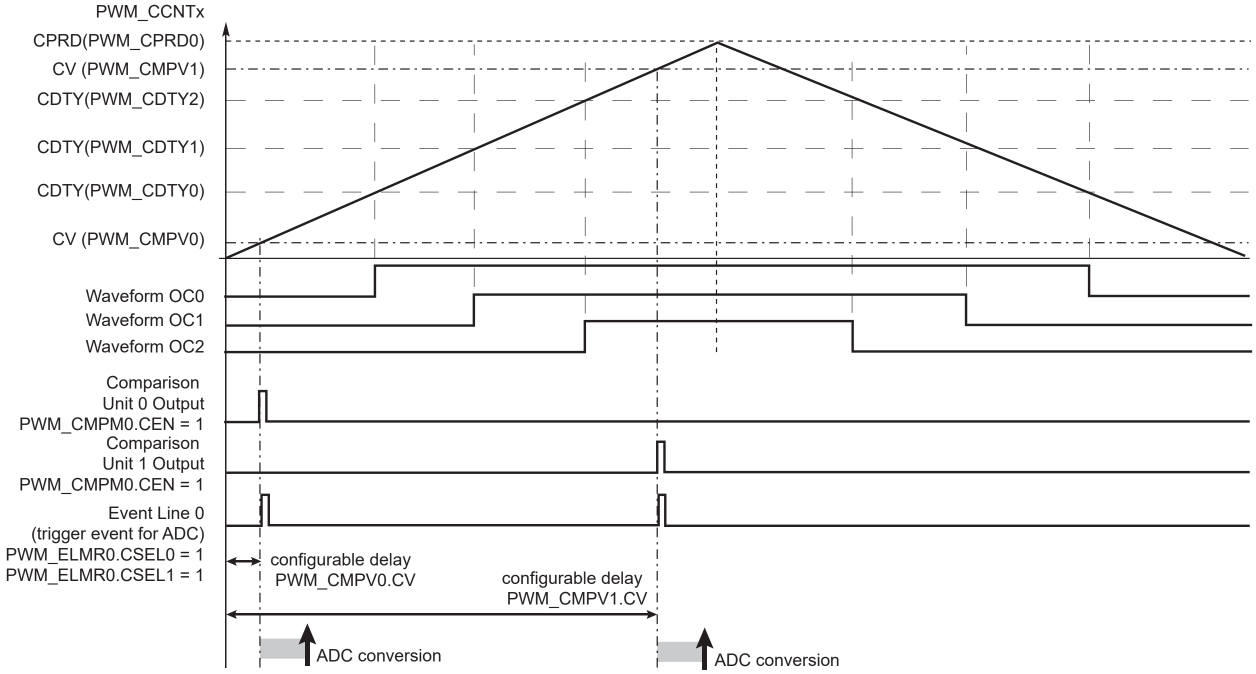


Figure 68-26. Event Line Generation Waveform (Example)



68.6.5 Debug Mode

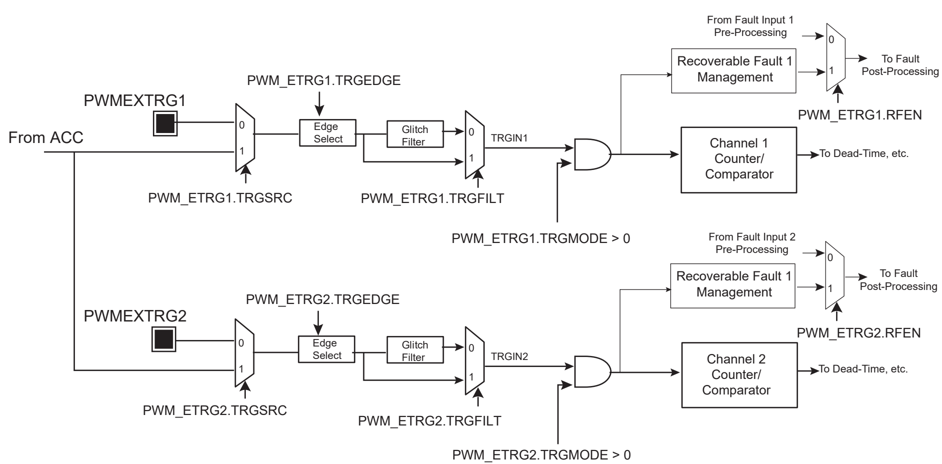
When debugging an application using breakpoints, the core is stopped in Debug mode when the breakpoint is hit. PWM registers are no longer updated by the application and thus PWML and/or PWMH outputs maintain the states or waveforms provided just before entering Debug mode. Depending on the application target, this state may not be valid for an undetermined period, or may have an undesired effect on devices driven by the PWM while the system is in Debug mode.

When entering Debug mode, the PWM outputs can be automatically and immediately forced to a predefined state if the bit OUTMODE=1 in the Debug register (PWM_DEBUG) (see PWM_DEBUG). The predefined state is defined in the Fault Protection Value registers (PWM_FPVx) (see PWM_FPV1 and PWM_FPV2).

68.6.6 PWM External Trigger Mode

The PWM channels 1 and 2 can be configured to use an external trigger for generating specific PWM signals to provide functions such as DC/DC converters, etc.

Figure 68-27. External Trigger Mode Block Diagram



The external trigger source can be selected through the bit TRGSRC of the PWM External Trigger Register (see the table below).

Table 68-6. External Event Source Selection

Channel	Trigger Source Selection	Trigger Source
1	PWM_ETRG1.TRGSRC = 0	From PWMEXTRG1 input
	PWM_ETRG1.TRGSRC = 1	From Analog Comparator Controller
2	PWM_ETRG2.TRGSRC = 0	From PWMEXTRG2 input
	PWM_ETRG2.TRGSRC = 1	From Analog Comparator Controller

Each external trigger source can be filtered by writing a one to PWM_ETRGx.TRGFILT.

When the external trigger event is detected, the internal counter of the PWM channel can be modified when conditions are met, depending on the value of the PWM_ETRGx.TRGMODE field.

Each time an external trigger event is detected, the corresponding PWM channel counter value is stored in PWM_ETRGx.MAXCNT if it is greater than the previously stored value. Reading PWM_ETRGx clears the MAXCNT value.

To adapt to different use cases, three different external trigger mode modes (ETM) are available for channels 1 and 2 depending on the value of the PWM_ETRGx.TRGMODE field:

- TRGMODE = 1: External PWM Reset Mode (see External PWM Reset Mode)
- TRGMODE = 2: External PWM Start Mode (see External PWM Start Mode)
- TRGMODE = 3: Cycle-By-Cycle Duty Mode (see Cycle-By-Cycle Duty Mode)

The ETM feature is disabled when PWM_ETRGx.TRGMODE = 0.

The ETM mode can be associated with the recoverable fault mode (PWM_ETRGx.RFEN=1) to manage specific use case (see Cycle-By-Cycle Duty Mode: LED String Control).

The use cases described in the figures External PWM Start Mode: Buck DC/DC Converter and Cycle-By-Cycle Duty Mode are managed by PWM_ETRGx.RFEN=0.

The ETM must be enabled only if the corresponding channel is left-aligned (CALG = 0 in PWM Channel Mode Register of channel 1 or 2) and not managed as a synchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register where x = 1 or 2). Programming the channel to be center-aligned or synchronous while TRGMODE is not 0 is forbidden.

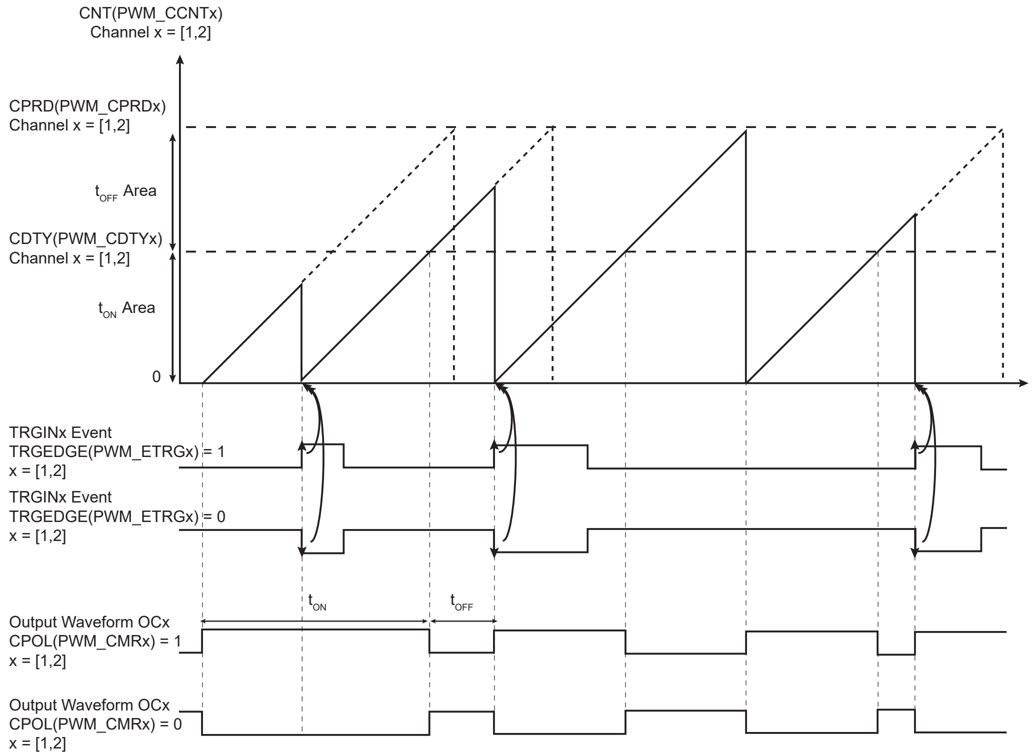
68.6.6.1 External PWM Reset Mode

External PWM Reset mode is selected by programming TRGMODE = 1 in the PWM_ETRGx register.

In this mode, when an edge is detected on the PWMEXTRGx input, the internal PWM counter is cleared and a new PWM cycle is restarted. The edge polarity can be selected by programming the TRGEDGE bit in the PWM_ETRGx register. If no trigger event is detected when the internal channel counter has reached the CPRD value in the PWM Channel Period Register, the internal counter is cleared and a new PWM cycle starts.

Note that this mode does not ensure a constant t_{ON} or t_{OFF} time.

Figure 68-28. External PWM Reset Mode



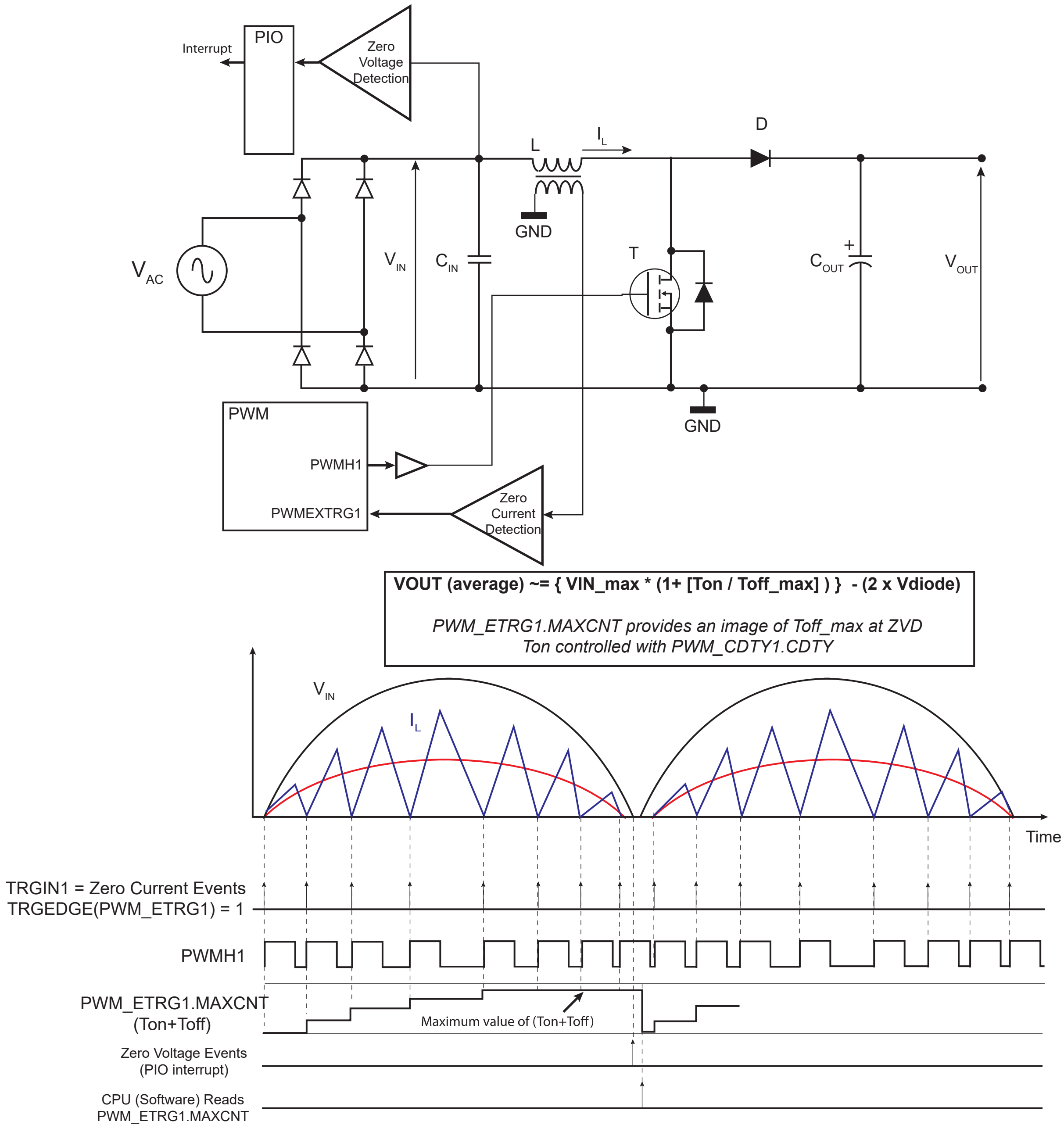
68.6.6.1.1 Application Example

The external PWM Reset mode can be used in power factor correction applications.

In the example below, the external trigger input is the PWMEXTRG1 (therefore the PWM channel used for regulation is the channel 1). The PWM channel 1 period (CPRD in the PWM Channel Period Register of the channel 1) must be programmed so that the TRGIN1 event always triggers before the PWM channel 1 period elapses.

In the figure below, an external circuit (not shown) is required to sense the inductor current I_L. The internal PWM counter of the channel 1 is cleared when the inductor current falls below a specific threshold (I_{REF}). This starts a new PWM period and increases the inductor current.

Figure 68-29. External PWM Reset Mode: Power Factor Correction Application



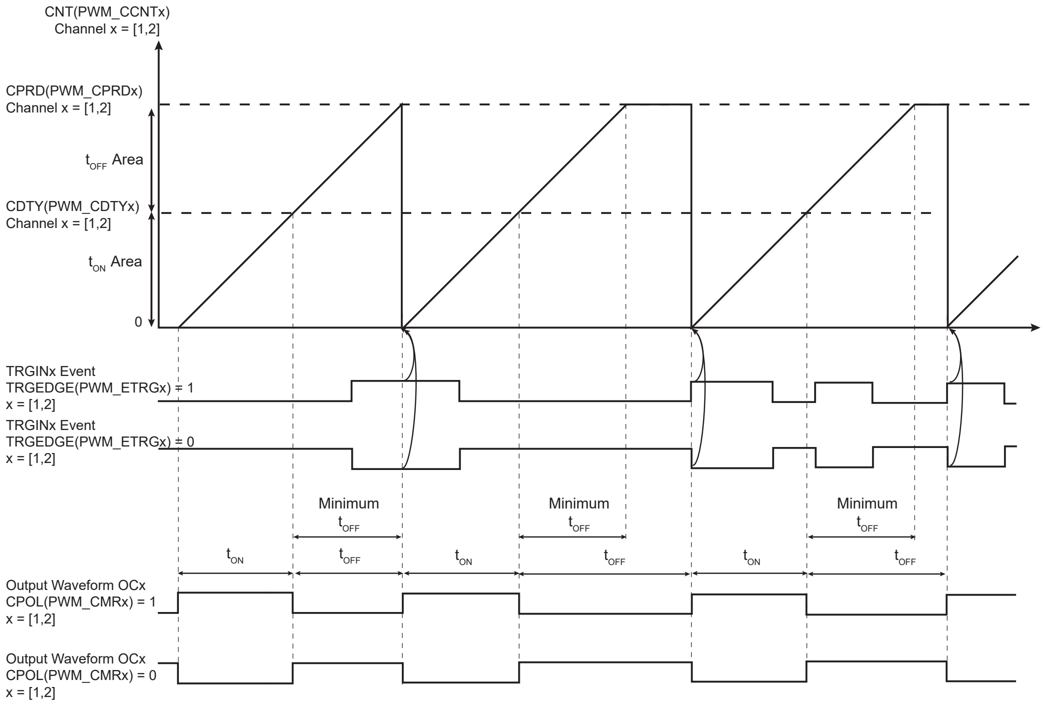
68.6.6.2 External PWM Start Mode

External PWM Start mode is selected by programming TRGMODE = 2 in the PWM_ETRGx register.

In this mode, the internal PWM counter can only be reset once it has reached the CPRD value in the PWM Channel Period Register and when the correct level is detected on the corresponding external trigger input. Both conditions have to be met to start a new PWM period. The active detection level is defined by the bit TRGEDGE of the PWM_ETRGx register.

Note that this mode ensures a constant t_{ON} time and a minimum t_{OFF} time.

Figure 68-30. External PWM Start Mode



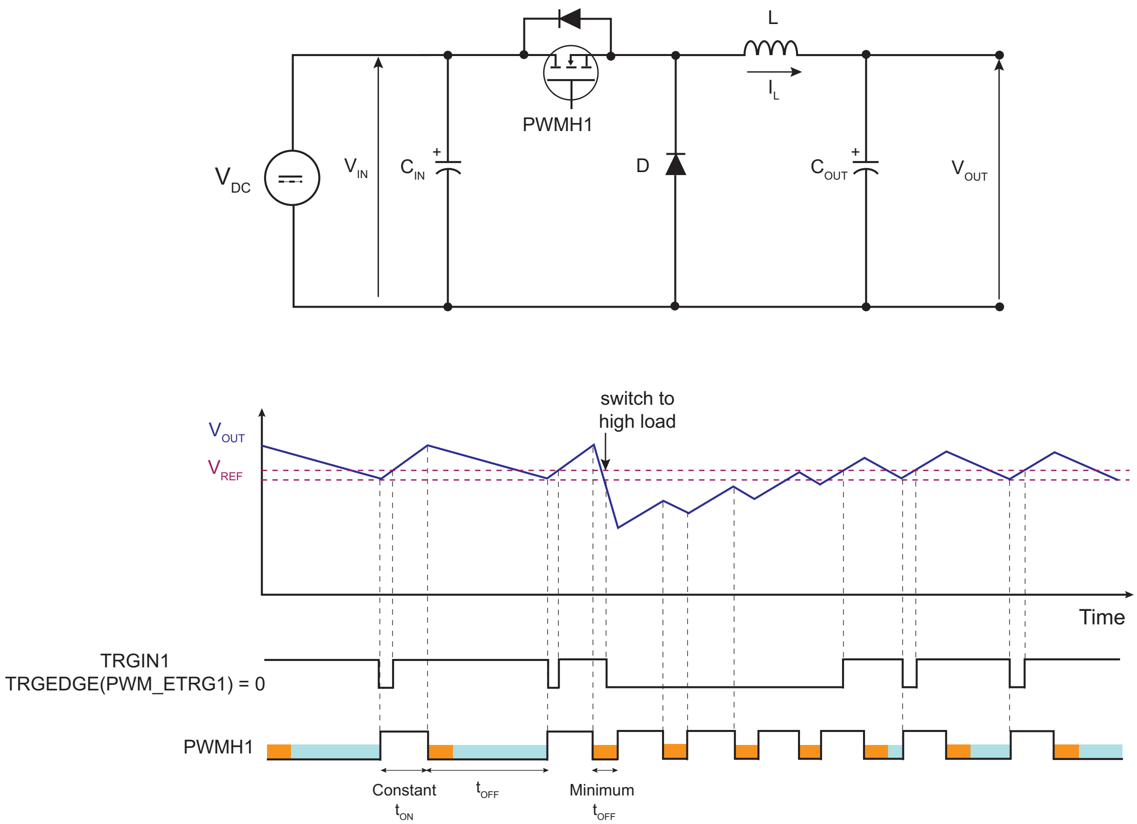
68.6.6.2.1 Application Example

The external PWM Start mode generates a modulated frequency PWM signal with a constant active level duration (t_{ON}) and a minimum inactive level duration (minimum t_{OFF}).

The t_{ON} time is defined by the CDTY value in the PWM Channel Duty Cycle Register. The minimum t_{OFF} time is defined by CDTY - CPRD (PWM Channel Period Register). This mode can be useful in Buck DC/DC Converter applications.

When the output voltage V_{OUT} is above a specific threshold (Vref), the PWM inactive level is maintained as long as V_{OUT} remains above this threshold. If V_{OUT} is below this specific threshold, this mode ensures a minimum t_{OFF} time required for MOSFET driving (see the figure below).

Figure 68-31. External PWM Start Mode: Buck DC/DC Converter



68.6.6.3 Cycle-By-Cycle Duty Mode

Cycle-by-cycle duty mode is selected by programming TRGMODE = 3 in PWM_ETRGx.

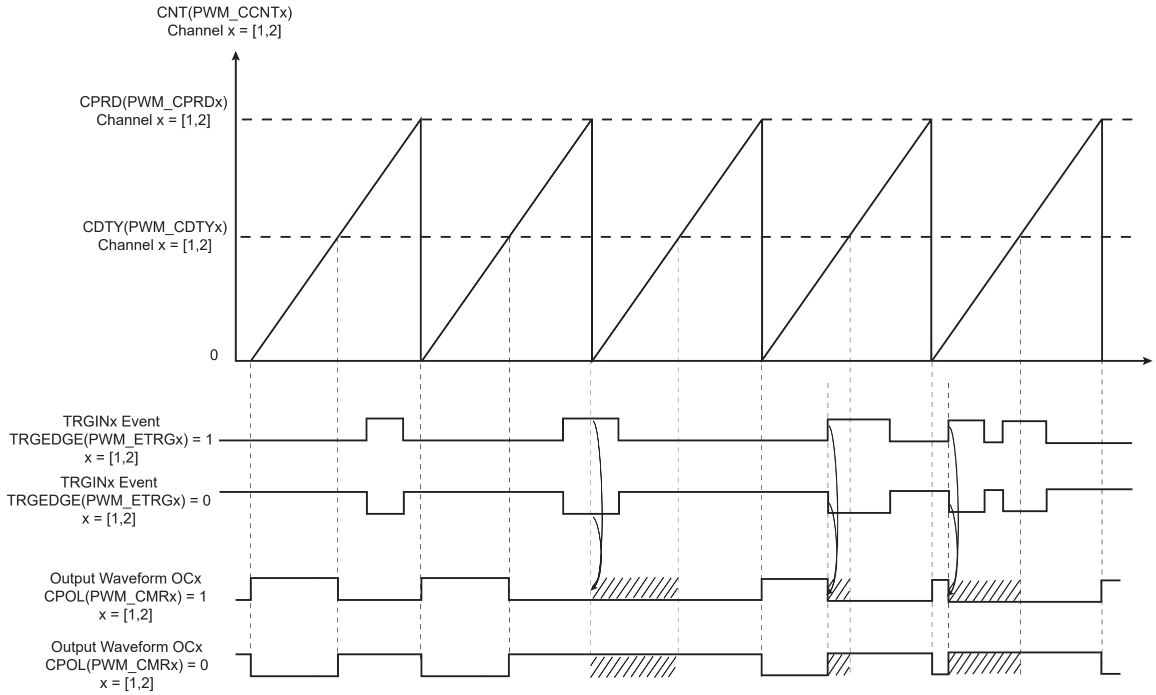
In this mode, the PWM frequency is constant and is defined by the CPRD value in the PWM Channel Period Register.

An external trigger event has no effect on the PWM output if it occurs while the internal PWM counter value is above the CDTY value of the PWM Channel Duty Cycle Register.

If the internal PWM counter value is below the value of CDTY of the PWM Channel Duty Cycle Register, an external trigger event makes the PWM output inactive.

The external trigger event can be detected on rising or falling edge according to the TRGEDGE bit in PWM_ETRGx.

Figure 68-32. Cycle-By-Cycle Duty Mode

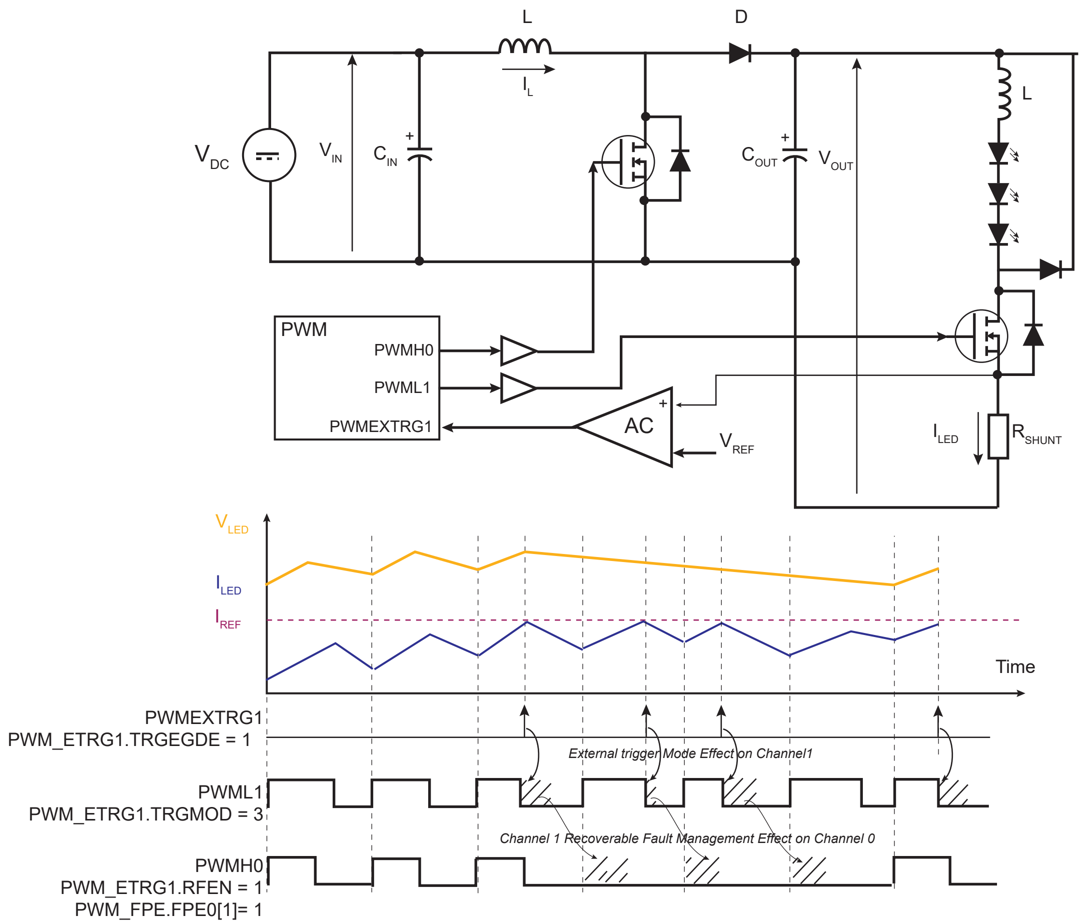


68.6.6.3.1 Application Example

The figure below illustrates an application example of the Cycle-by-cycle Duty mode.

In an LED string control circuit, Cycle-by-cycle Duty mode can be used to automatically limit the current in the LED string. This use case requires the recoverable fault mode to be enabled on channel 1 and associated fault signal must be enabled on channel 0.

Figure 68-33. Cycle-By-Cycle Duty Mode: LED String Control



68.6.6.4 Leading-Edge Blanking (LEB)

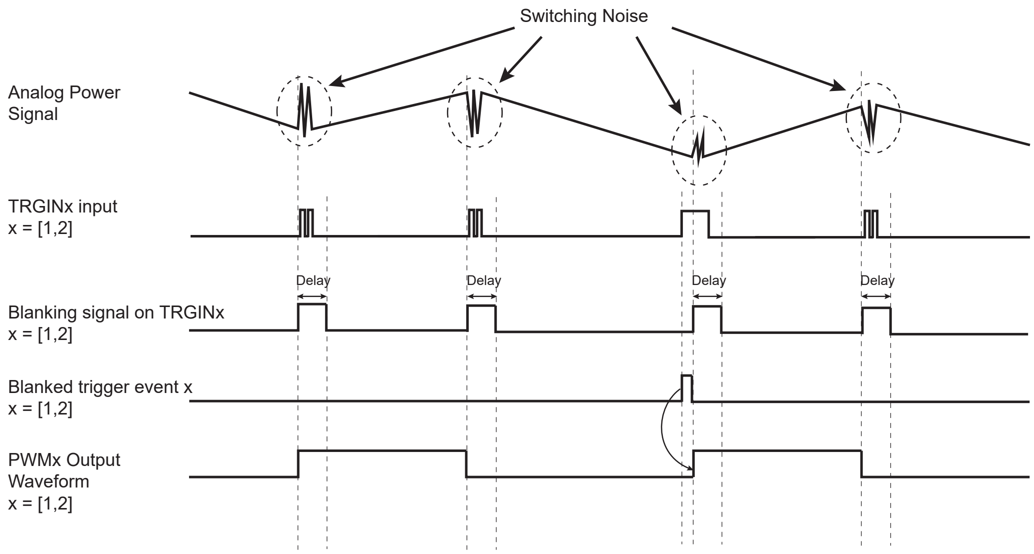
PWM channels 1 and 2 support leading-edge blanking. Leading-edge blanking masks the external trigger input when a transient occurs on the corresponding PWM output. It masks potential spurious external events due to power transistor switching.

The blanking delay on each external trigger input is configured by programming the LEBDELAYx in the PWM Leading-Edge Blanking Register.

The LEB can be enabled on both the rising and the falling edges for the PWMH and PWML outputs through the bits PWMLFEN, PWMLREN, PWMHFEN, PWMHREN.

Any event on the PWMEXTRGx input which occurs during the blanking time is ignored.

Figure 68-34. Leading-Edge Blanking



68.6.7 PWM Controller Operations

68.6.7.1 Initialization

Before enabling the channels, they must be configured by the software application as described below:

- Unlock User Interface by writing the WPCMD field in PWM_WPCR.
- Configuration of the clock generator (DIVA, PREA, DIVB, PREB in the PWM_CLK register if required).
- Selection of the clock for each channel (CPRE field in PWM_CMRx)
- Configuration of the waveform alignment for each channel (CALG field in PWM_CMRx)
- Selection of the counter event selection (if CALG = 1) for each channel (CES field in PWM_CMRx)
- Configuration of the output waveform polarity for each channel (CPOL bit in PWM_CMRx)
- Configuration of the period for each channel (CPRD in the PWM_CPRDx register). Writing in PWM_CPRDx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CPRDUPDx register to update PWM_CPRDx as explained below.
- Configuration of the duty-cycle for each channel (CDTY in the PWM_CDTYx register). Writing in PWM_CDTYx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CDTYUPDx register to update PWM_CDTYx as explained below.
- Configuration of the dead-time generator for each channel (DTH and DTL in PWM_DTx) if enabled (DTE bit in PWM_CMRx). Writing in the PWM_DTx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_DTUPDx register to update PWM_DTx
- Selection of the synchronous channels (SYNCx in the PWM_SCM register)
- Selection of the moment when the WRDY flag and the corresponding DMA Controller transfer request are set (PTRM and PTRCS in the PWM_SCM register)
- Configuration of the Update mode (UPDM in PWM_SCM register)
- Configuration of the update period (UPR in PWM_SCUP register) if needed
- Configuration of the comparisons (PWM_CMPVx and PWM_CMPMx)
- Configuration of the event lines (PWM_ELMRx)
- Configuration of the fault inputs polarity (FPOL in PWM_FMR)
- Configuration of the fault protection (FMOD and FFIL in PWM_FMR, PWM_FPV and PWM_FPE1)
- Enable of the interrupts (writing CHIDx and FCHIDx in PWM_IER1, and writing WRDY, UNRE, CMPMx and CMPUx in PWM_IER2)
- Enable of the PWM channels (writing CHIDx in the PWM_ENA register)

68.6.7.2 Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the PWM Channel Period Register (PWM_CPRDx) and the PWM Channel Duty Cycle Register (PWM_CDTYx) helps the user select the appropriate clock. The event number written in the Period Register gives the PWM accuracy. The Duty-Cycle quantum cannot be lower than 1/CPRDx value. The higher the value of PWM_CPRDx, the greater the PWM accuracy.

For example, if the user sets 15 (in decimal) in PWM_CPRDx, the user is able to set a value from between 1 up to 14 in PWM_CDTYx. The resulting duty-cycle quantum cannot be lower than 1/15 of the PWM period.

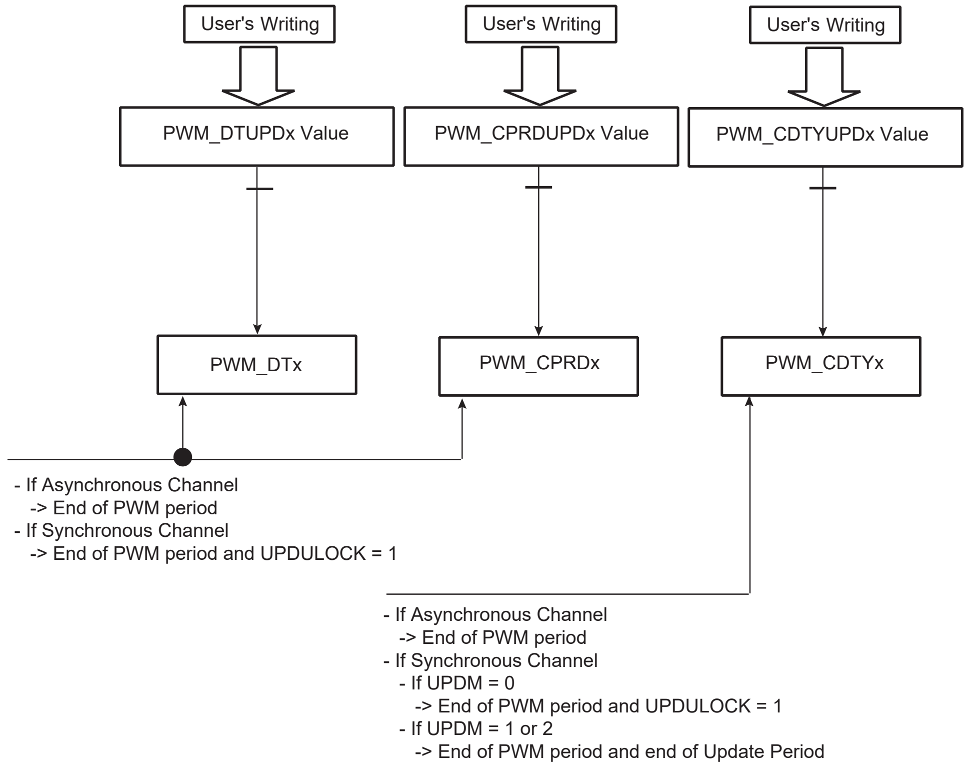
68.6.7.3 Changing the Duty-Cycle, the Period and the Dead-Times

It is possible to modulate the output waveform duty-cycle, period and dead-times.

To prevent unexpected output waveform, the user must use the PWM Channel Duty Cycle Update Register (PWM_CDTYUPDx), the PWM Channel Period Update Register (PWM_CPRDUPDx) and the PWM Channel Dead Time Update Register (PWM_DTUPDx) to change waveform parameters while the channel is still enabled.

- If the channel is an asynchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register (PWM_SCM)), these registers hold the new period, duty-cycle and dead-times values until the end of the current PWM period and update the values for the next period.
- If the channel is a synchronous channel and update method 0 is selected (SYNCx = 1 and UPDM = 0 in PWM_SCM register), these registers hold the new period, duty-cycle and dead-times values until the bit UPDULOCK is written at ‘1’ (in PWM Sync Channels Update Control Register (PWM_SCUC)) and the end of the current PWM period, then update the values for the next period.
- If the channel is a synchronous channel and update method 1 or 2 is selected (SYNCx = 1 and UPDM = 1 or 2 in PWM_SCM register):
  – registers PWM_CPRDUPDx and PWM_DTUPDx hold the new period and dead-times values until the bit UPDULOCK is written at ‘1’ (in PWM_SCUC) and the end of the current PWM period, then update the values for the next period.
  – register PWM_CDTYUPDx holds the new duty-cycle value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM Sync Channels Update Period Register (PWM_SCUP)) and the end of the current PWM period, then updates the value for the next period.
  Note: If the update registers PWM_CDTYUPDx, PWM_CPRDUPDx and PWM_DTUPDx are written several times between two updates, only the last written value is taken into account.

Figure 68-35. Synchronized Period, Duty-Cycle and Dead-Time Update



68.6.7.4 Changing the Update Period of Synchronous Channels

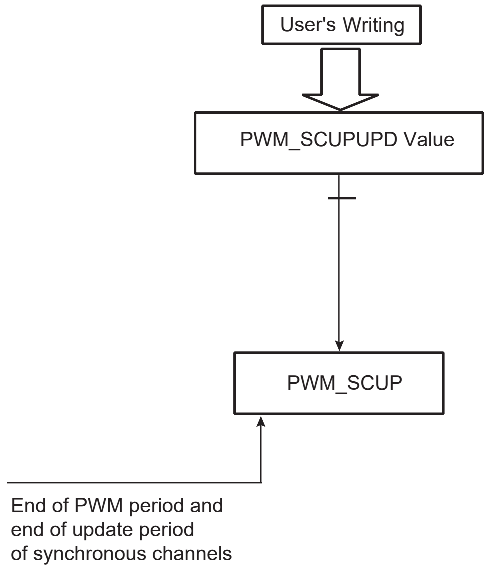
It is possible to change the update period of synchronous channels while they are enabled. See Method 2: Manual write of duty-cycle values and automatic trigger of the update and Method 3: Automatic write of duty-cycle values and automatic trigger of the update .

To prevent an unexpected update of the synchronous channels registers, the user must use the PWM Sync Channels Update Period Update Register (PWM_SCUPUPD) to change the update period of synchronous channels while they are still enabled. This register holds the new value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM_SCUP) and the end of the current PWM period, then updates the value for the next period.

Notes:

1. If the update register PWM_SCUPUPD is written several times between two updates, only the last written value is taken into account.
2. Changing the update period does make sense only if there is one or more synchronous channels and if the update method 1 or 2 is selected (UPDM = 1 or 2 in PWM Sync Channels Mode Register).

Figure 68-36. Synchronized Update of Update Period Value of Synchronous Channels



68.6.7.5 Changing the Comparison Value and the Comparison Configuration

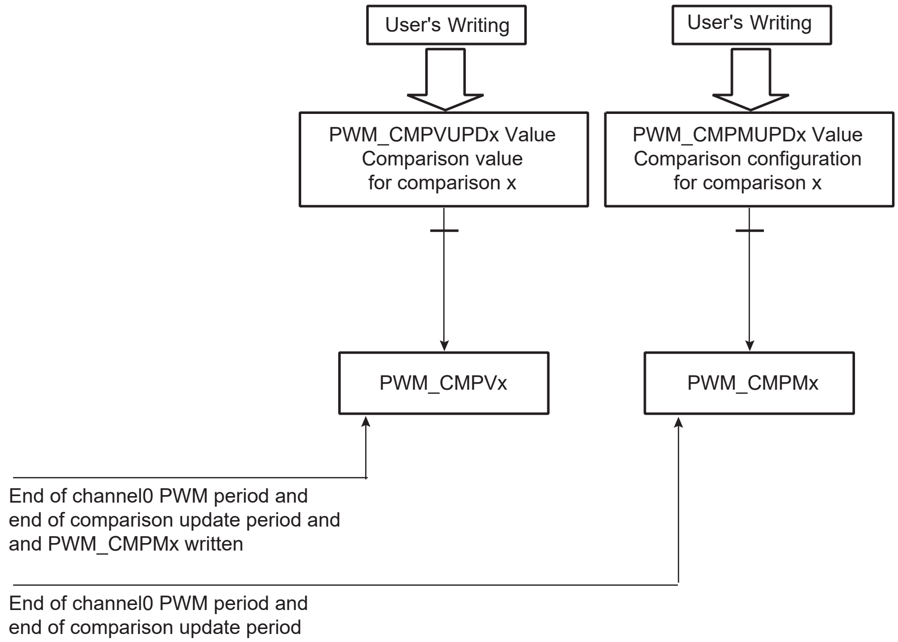
It is possible to change the comparison values and the comparison configurations while the channel 0 is enabled (see PWM Comparison Units).

To prevent unexpected comparison match, the user must use the PWM Comparison x Value Update Register (PWM_CMPVUPDx) and the PWM Comparison x Mode Update Register (PWM_CMPMUPDx) to change, respectively, the comparison values and the comparison configurations while the channel 0 is still enabled. These registers hold the new values until the end of the comparison update period (when CUPRCNT is equal to CUPR in PWM Comparison x Mode Register (PWM_CMPMx) and the end of the current PWM period, then update the values for the next period.

⚠ CAUTION The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

Note: If the update registers PWM_CMPVUPDx and PWM_CMPMUPDx are written several times between two updates, only the last written value are taken into account.

Figure 68-37. Synchronized Update of Comparison Values and Configurations



68.6.7.6 Interrupt Sources

Depending on the interrupt mask in PWM_IMR1 and PWM_IMR2, an interrupt can be generated at the end of the corresponding channel period (CHIDx in the PWM Interrupt Status Register 1 (PWM_ISR1)), after a fault event (FCHIDx in PWM_ISR1), after a comparison match (CMPMx in PWM_ISR2), after a comparison update (CMPUx in PWM_ISR2) or according to the Transfer mode of the synchronous channels (WRDY and UNRE in PWM_ISR2).

If the interrupt is generated by the flags CHIDx or FCHIDx, the interrupt remains active until a read operation in PWM_ISR1 occurs.

If the interrupt is generated by the flags WRDY or UNRE or CMPMx or CMPUx, the interrupt remains active until a read operation in PWM_ISR2 occurs.

A channel interrupt is enabled by setting the corresponding bit in PWM_IER1 and PWM_IER2. A channel interrupt is disabled by setting the corresponding bit in PWM_IDR1 and PWM_IDR2.

68.6.8 Register Write Protection

To prevent any single software error that may corrupt PWM behavior, the registers listed below can be write-protected by writing the field WPCMD in the PWM Write Protection Control Register (PWM_WPCR). They are divided into six groups:

- Register group 0:
  – PWM Clock Register
- Register group 1:
  – PWM Disable Register
  – PWM Interrupt Enable Register 1
  – PWM Interrupt Disable Register 1
  – PWM Interrupt Enable Register 2
  – PWM Interrupt Disable Register 2
- Register group 2:
  – PWM Sync Channels Mode Register
  – PWM Channel Mode Register
  – PWM Stepper Motor Mode Register
  – PWM Fault Protection Value Register 2
  – PWM Leading-Edge Blanking Register

- PWM Channel Mode Update Register
- Register group 3:
  - PWM Spread Spectrum Register
  - PWM Spread Spectrum Update Register
  - PWM Channel Period Register
  - PWM Channel Period Update Register
- Register group 4:
  - PWM Channel Dead Time Register
  - PWM Channel Dead Time Update Register
- Register group 5:
  - PWM Fault Mode Register
  - PWM Fault Protection Value Register 1

There are two types of write protection:

- SW write protection—can be enabled or disabled by software
- HW write protection—can be enabled by software but only disabled by a hardware reset of the PWM controller

Both types of write protection can be applied independently to a particular register group by means of the WPCMD and WPRGx fields in PWM_WPCR. If at least one type of write protection is active, the register group is write-protected. The value of field WPCMD defines the action to be performed:

- 0: Disables SW write protection of the register groups of which the bit WPRGx is at ‘1’
- 1: Enables SW write protection of the register groups of which the bit WPRGx is at ‘1’
- 2: Enables HW write protection of the register groups of which the bit WPRGx is at ‘1’

At any time, the user can determine whether SW or HW write protection is active in a particular register group by the fields WPSWS and WPHWS in the PWM Write Protection Status Register (PWM_WPSR).

If a write access to a write-protected register is detected, the WPVS flag in PWM_WPSR is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS and WPVSRC fields are automatically cleared after reading PWM_WPSR.

68.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PWM_CLK	31:24					PREB[3:0]			
		23:16	DIVB[7:0]							
		15:8					PREA[3:0]			
		7:0	DIVA[7:0]							
0x04	PWM_ENA	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x08	PWM_DIS	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x0C	PWM_SR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x10	PWM_IER1	31:24								
		23:16					FCHID3	FCHID2	FCHID1	FCHID0
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x14	PWM_IDR1	31:24								
		23:16					FCHID3	FCHID2	FCHID1	FCHID0
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x18	PWM_IMR1	31:24								
		23:16					FCHID3	FCHID2	FCHID1	FCHID0
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x1C	PWM_ISR1	31:24								
		23:16					FCHID3	FCHID2	FCHID1	FCHID0
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x20	PWM_SCM	31:24								
		23:16	PTRCS[2:0]			PTRM			UPDM[1:0]	
		15:8								
		7:0					SYNC3	SYNC2	SYNC1	SYNC0
0x24	PWM_DMAR	31:24								
		23:16	DMADUTY[23:16]							
		15:8	DMADUTY[15:8]							
		7:0	DMADUTY[7:0]							
0x28	PWM_SCUC	31:24								
		23:16								
		15:8								
		7:0								UPDULOCK
0x2C	PWM_SCUP	31:24								
		23:16								
		15:8								
		7:0	UPRCNT[3:0]				UPR[3:0]			
0x30	PWM_SCUPUPD	31:24								
		23:16								
		15:8								
		7:0					UPRUPD[3:0]			
0x34	PWM_IER2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE			WRDY

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	PWM_IDR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE			WRDY
0x3C	PWM_IMR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE			WRDY
0x40	PWM_ISR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE			WRDY
0x44	PWM_OOV	31:24								
		23:16					OOVL3	OOVL2	OOVL1	OOVL0
		15:8								
		7:0					OOVH3	OOVH2	OOVH1	OOVH0
0x48	PWM_OS	31:24								
		23:16					OSL3	OSL2	OSL1	OSL0
		15:8								
		7:0					OSH3	OSH2	OSH1	OSH0
0x4C	PWM_OSS	31:24								
		23:16					OSSL3	OSSL2	OSSL1	OSSL0
		15:8								
		7:0					OSSH3	OSSH2	OSSH1	OSSH0
0x50	PWM_OSC	31:24								
		23:16					OSCL3	OSCL2	OSCL1	OSCL0
		15:8								
		7:0					OSCH3	OSCH2	OSCH1	OSCH0
0x54	PWM_OSSUPD	31:24								
		23:16					OSSUPL3	OSSUPL2	OSSUPL1	OSSUPL0
		15:8								
		7:0					OSSUPH3	OSSUPH2	OSSUPH1	OSSUPH0
0x58	PWM_OSCUPD	31:24								
		23:16					OSCUPL3	OSCUPL2	OSCUPL1	OSCUPL0
		15:8								
		7:0					OSCUPH3	OSCUPH2	OSCUPH1	OSCUPH0
0x5C	PWM_FMR	31:24								
		23:16	FFIL[7:0]							
		15:8	FMOD[7:0]							
		7:0	FPOL[7:0]							
0x60	PWM_FSR	31:24								
		23:16								
		15:8	FS[7:0]							
		7:0	FIV[7:0]							
0x64	PWM_FCR	31:24								
		23:16								
		15:8								
		7:0	FCLR[7:0]							
0x68	PWM_FPV1	31:24								
		23:16					FPVL3	FPVL2	FPVL1	FPVL0
		15:8								
		7:0					FPVH3	FPVH2	FPVH1	FPVH0
0x6C	PWM_FPE	31:24	FPE3[7:0]							
		23:16	FPE2[7:0]							
		15:8	FPE1[7:0]							
		7:0	FPE0[7:0]							
0x70 ... 0x7B	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x7C	PWM_ELMR0	31:24								
		23:16								
		15:8								
		7:0	CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0
0x80	PWM_ELMR1	31:24								
		23:16								
		15:8								
		7:0	CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0
0x84 ... 0x9F	Reserved									
0xA0	PWM_SSPR	31:24								SPRDM
		23:16	SPRD[23:16]							
		15:8	SPRD[15:8]							
		7:0	SPRD[7:0]							
0xA4	PWM_SSPUP	31:24								
		23:16	SPRDUP[23:16]							
		15:8	SPRDUP[15:8]							
		7:0	SPRDUP[7:0]							
0xA8 ... 0xAB	Reserved									
0xAC	PWM_DEBUG	31:24								
		23:16								
		15:8								
		7:0								OUTMODE
0xB0	PWM_SMMR	31:24								
		23:16							DOWN1	DOWN0
		15:8								
		7:0							GCEN1	GCEN0
0xB4 ... 0xBF	Reserved									
0xC0	PWM_FPV2	31:24								
		23:16					FPZL3	FPZL2	FPZL1	FPZL0
		15:8								
		7:0					FPZH3	FPZH2	FPZH1	FPZH0
0xC4 ... 0xE3	Reserved									
0xE4	PWM_WPCR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD[1:0]	
0xE8	PWM_WPSR	31:24	WPVSRC[15:8]							
		23:16	WPVSRC[7:0]							
		15:8			WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0
		7:0	WPVS		WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0
0xEC ... 0x012F	Reserved									
0x0130	PWM_CMPV0	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0134	PWM_CMPVUPD0	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0138	PWM_CMPM0	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x013C	PWM_CMPMUPD0	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0140	PWM_CMPV1	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0144	PWM_CMPVUPD1	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0148	PWM_CMPM1	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x014C	PWM_CMPMUPD1	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0150	PWM_CMPV2	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0154	PWM_CMPVUPD2	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0158	PWM_CMPM2	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x015C	PWM_CMPMUPD2	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0160	PWM_CMPV3	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0164	PWM_CMPVUPD3	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0168	PWM_CMPM3	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x016C	PWM_CMPMUPD3	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0170	PWM_CMPV4	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0174	PWM_CMPVUPD4	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0178	PWM_CMPM4	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x017C	PWM_CMPMUPD4	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0180	PWM_CMPV5	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0184	PWM_CMPVUPD5	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0188	PWM_CMPM5	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x018C	PWM_CMPMUPD5	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0190	PWM_CMPV6	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0194	PWM_CMPVUPD6	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0198	PWM_CMPM6	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x019C	PWM_CMPMUPD6	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x01A0	PWM_CMPV7	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x01A4	PWM_CMPVUPD7	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01A8	PWM_CMPM7	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x01AC	PWM_CMPMUPD7	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x01B0 ... 0x01FF	Reserved									
0x0200	PWM_CMR0	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0204	PWM_CDTY0	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0208	PWM_CDTYUPD0	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x020C	PWM_CPRD0	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0210	PWM_CPRDUPD0	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0214	PWM_CCNT0	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0218	PWM_DT0	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x021C	PWM_DTUPD0	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0220	PWM_CMR1	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0224	PWM_CDTY1	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0228	PWM_CDTYUPD1	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x022C	PWM_CPRD1	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0230	PWM_CPRDUPD1	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0234	PWM_CCNT1	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0238	PWM_DT1	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x023C	PWM_DTUPD1	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0240	PWM_CMR2	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0244	PWM_CDTY2	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0248	PWM_CDTYUPD2	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x024C	PWM_CPRD2	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0250	PWM_CPRDUPD2	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0254	PWM_CCNT2	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0258	PWM_DT2	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x025C	PWM_DTUPD2	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0260	PWM_CMR3	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0264	PWM_CDTY3	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0268	PWM_CDTYUPD3	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x026C	PWM_CPRD3	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0270	PWM_CPRDUPD3	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0274	PWM_CCNT3	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0278	PWM_DT3	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x027C	PWM_DTUPD3	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0280 ... 0x03FF	Reserved									
0x0400	PWM_CMUPD0	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0404 ... 0x041F	Reserved									
0x0420	PWM_CMUPD1	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0424 ... 0x042B	Reserved									
0x042C	PWM_ETRG1	31:24	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
		23:16	MAXCNT[23:16]							
		15:8	MAXCNT[15:8]							
		7:0	MAXCNT[7:0]							
0x0430	PWM_LEBR1	31:24								
		23:16					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
		15:8								
		7:0		LEBDELAY[6:0]						
0x0434 ... 0x043F	Reserved									
0x0440	PWM_CMUPD2	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0444 ... 0x044B	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x044C	PWM_ETRG2	31:24	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
		23:16	MAXCNT[23:16]							
		15:8	MAXCNT[15:8]							
		7:0	MAXCNT[7:0]							
0x0450	PWM_LEBR2	31:24								
		23:16					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
		15:8								
		7:0		LEBDELAY[6:0]						
0x0454 ... 0x045F	Reserved									
0x0460	PWM_CMUPD3	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								

68.7.1 PWM Clock Register

Name: PWM_CLK
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS0 and WPHWS0 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
					PREB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIVB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PREA[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIVA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:24 – PREB[3:0] CLKB Source Clock Selection

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

Bits 23:16 – DIVB[7:0] CLKB Divide Factor

Value	Name	Description
0	CLKB_POFF	CLKB clock is turned off
1	PREB	CLKB clock is clock selected by PREB
2-255	PREB_DIV	CLKB clock is clock selected by PREB divided by DIVB factor

Bits 11:8 – PREA[3:0] CLKA Source Clock Selection

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4

Value	Name	Description
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

Bits 7:0 – DIVA[7:0] CLKA Divide Factor

Value	Name	Description
0	CLKA_POFF	CLKA clock is turned off
1	PREA	CLKA clock is clock selected by PREA
2-255	PREA_DIV	CLKA clock is clock selected by PREA divided by DIVA factor

68.7.2 PWM Enable Register

Name: PWM_ENA
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Enable PWM output for channel x.

68.7.3 PWM Disable Register

Name: PWM_DIS
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Disable PWM output for channel x.

68.7.4 PWM Status Register

Name: PWM_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	PWM output for channel x is disabled.
1	PWM output for channel x is enabled.

68.7.5 PWM Interrupt Enable Register 1

Name: PWM_IER1
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x Interrupt Enable

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x Interrupt Enable

68.7.6 PWM Interrupt Disable Register 1

Name: PWM_IDR1
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x Interrupt Disable

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x Interrupt Disable

68.7.7 PWM Interrupt Mask Register 1

Name: PWM_IMR1
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x Interrupt Mask

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x Interrupt Mask

68.7.8 PWM Interrupt Status Register 1

Name: PWM_ISR1
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Note: Reading PWM_ISR1 automatically clears CHIDx and FCHIDx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x

Value	Description
0	No new trigger of the fault protection since the last read of PWM_ISR1.
1	At least one trigger of the fault protection since the last read of PWM_ISR1.

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x

Value	Description
0	No new counter event has occurred since the last read of PWM_ISR1.
1	At least one counter event has occurred since the last read of PWM_ISR1.

68.7.9 PWM Sync Channels Mode Register

Name: PWM_SCM
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PTRCS[2:0]			PTRM			UPDM[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					SYNC3	SYNC2	SYNC1	SYNC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 23:21 – PTRCS[2:0] DMA Controller Transfer Request Comparison Selection
Selection of the comparison used to set the flag WRDY and the corresponding DMA Controller transfer request.

Bit 20 – PTRM DMA Controller Transfer Request Mode

UPDM	PTRM	WRDY Flag and DMA Controller Transfer Request
0	x	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are never set to '1'.
1	x	The WRDY flag in PWM Interrupt Status Register 2 is set to '1' as soon as the update period is elapsed, the DMA Controller transfer request is never set to '1'.
2	0	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are set to '1' as soon as the update period is elapsed.
	1	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are set to '1' as soon as the selected comparison matches.

Bits 17:16 – UPDM[1:0] Synchronous Channels Update Mode

Value	Name	Description
0	MODE0	Manual write of double buffer registers and manual update of synchronous channels 1[(1)].
1	MODE1	Manual write of double buffer registers and automatic update of synchronous channels 2[(2)].
2	MODE2	The WRDY flag in PWM Interrupt Status Register 2 and the DMA transfer request are set to '1' as soon as the update period is elapsed.

Notes:
1. The update occurs at the beginning of the next PWM period, when the UPDULOCK bit in PWM Sync Channels Update Control Register is set.
2. The update occurs when the Update Period is elapsed.

Bits 0, 1, 2, 3 – SYNCx Synchronous Channel x

Value	Description
0	Channel x is not a synchronous channel.
1	Channel x is a synchronous channel.

68.7.10 PWM DMA Register

Name: PWM_DMAR
Offset: 0x24
Reset: –
Property: Write-only

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	DMADUTY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	DMADUTY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	DMADUTY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – DMADUTY[23:0] Duty-Cycle Holding Register for DMA Access
Each write access to PWM_DMAR sequentially updates PWM_CDTYUPDx.CDTYUPD with DMADUTY (only for channel configured as synchronous). See “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” .

68.7.11 PWM Sync Channels Update Control Register

Name: PWM_SCUC
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								UPDULOCK
Access								R/W
Reset								0

Bit 0 – UPDULOCK Synchronous Channels Update Unlock
This bit is automatically reset when the update is done.

Value	Description
0	No effect
1	If the UPDM field is set to ‘0’ in PWM Sync Channels Mode Register, writing the UPDULOCK bit to ‘1’ triggers the update of the period value, the duty-cycle and the dead-time values of synchronous channels at the beginning of the next PWM period. If the field UPDM is set to ‘1’ or ‘2’, writing the UPDULOCK bit to ‘1’ triggers only the update of the period value and of the dead-time values of synchronous channels.

68.7.12 PWM Sync Channels Update Period Register

Name: PWM_SCUP
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	UPRCNT[3:0]				UPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – UPRCNT[3:0] Update Period Counter
Reports the value of the update period counter.

Bits 3:0 – UPR[3:0] Update Period
Defines the time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

68.7.13 PWM Sync Channels Update Period Update Register

Name: PWM_SCUPUPD
Offset: 0x30
Reset: –
Property: Write-only

This register acts as a double buffer for the UPR value. This prevents an unexpected automatic trigger of the update of synchronous channels.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					UPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bits 3:0 – UPRUPD[3:0] Update Period Update
Defines the wanted time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

68.7.14 PWM Interrupt Enable Register 2

Name: PWM_IER2
Offset: 0x34
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
					UNRE			WRDY
Access					W			W
Reset					–			–

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Enable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Enable

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Enable

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Enable

68.7.15 PWM Interrupt Disable Register 2

Name: PWM_IDR2
Offset: 0x38
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
					UNRE			WRDY
Access					W			W
Reset					–			–

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Disable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Disable

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Disable

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Disable

68.7.16 PWM Interrupt Mask Register 2

Name: PWM_IMR2
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					UNRE			WRDY
Access					R			R
Reset					0			0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Mask

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Mask

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Mask

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Mask

68.7.17 PWM Interrupt Status Register 2

Name: PWM_ISR2
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Reading PWM_ISR2 automatically clears flags WRDY, UNRE and CMPSx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					UNRE			WRDY
Access					R			R
Reset					0			0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update

Value	Description
0	The comparison x has not been updated since the last read of the PWM_ISR2 register.
1	The comparison x has been updated at least one time since the last read of the PWM_ISR2 register.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match

Value	Description
0	The comparison x has not matched since the last read of the PWM_ISR2 register.
1	The comparison x has matched at least one time since the last read of the PWM_ISR2 register.

Bit 3 – UNRE Synchronous Channels Update Underrun Error

Value	Description
0	No Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.
1	At least one Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.

Bit 0 – WRDY Write Ready for Synchronous Channels Update

Value	Description
0	New duty-cycle and dead-time values for the synchronous channels cannot be written.
1	New duty-cycle and dead-time values for the synchronous channels can be written.

68.7.18 PWM Output Override Value Register

Name: PWM_OOV
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OOVL3	OOVL2	OOVL1	OOVL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OOVH3	OOVH2	OOVH1	OOVH0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – OOVLx Output Override Value for PWML output of the channel x

Value	Description
0	Override value is 0 for PWML output of channel x.
1	Override value is 1 for PWML output of channel x.

Bits 0, 1, 2, 3 – OOVHx Output Override Value for PWMH output of the channel x

Value	Description
0	Override value is 0 for PWMH output of channel x.
1	Override value is 1 for PWMH output of channel x.

68.7.19 PWM Output Selection Register

Name: PWM_OS
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OSL3	OSL2	OSL1	OSL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OSH3	OSH2	OSH1	OSH0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – OSLx Output Selection for PWML output of the channel x

Value	Description
0	Dead-time generator output DTOLx selected as PWML output of channel x.
1	Output override value OOVLx selected as PWML output of channel x.

Bits 0, 1, 2, 3 – OSHx Output Selection for PWMH output of the channel x

Value	Description
0	Dead-time generator output DTOHx selected as PWMH output of channel x.
1	Output override value OOVHx selected as PWMH output of channel x.

68.7.20 PWM Output Selection Set Register

Name: PWM_OSS
Offset: 0x4C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
					OSSL3	OSSL2	OSSL1	OSSL0
Access					W	W	W	W
Reset					–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					OSSH3	OSSH2	OSSH1	OSSH0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – OSSLx Output Selection Set for PWML output of the channel x

Value	Description
0	No effect.
1	Output override value OOVLx selected as PWML output of channel x.

Bits 0, 1, 2, 3 – OSSHx Output Selection Set for PWMH output of the channel x

Value	Description
0	No effect.
1	Output override value OOVHx selected as PWMH output of channel x.

68.7.21 PWM Output Selection Clear Register

Name: PWM_OSC
Offset: 0x50
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OSCL3	OSCL2	OSCL1	OSCL0
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OSCH3	OSCH2	OSCH1	OSCH0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – OSCLx Output Selection Clear for PWML output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOLx selected as PWML output of channel x.

Bits 0, 1, 2, 3 – OSCHx Output Selection Clear for PWMH output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOHx selected as PWMH output of channel x.

68.7.22 PWM Output Selection Set Update Register

Name: PWM_OSSUPD
Offset: 0x54
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OSSUPL3	OSSUPL2	OSSUPL1	OSSUPL0
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OSSUPH3	OSSUPH2	OSSUPH1	OSSUPH0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – OSSUPLx Output Selection Set for PWML output of the channel x

Value	Description
0	No effect.
1	Output override value OOVLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

Bits 0, 1, 2, 3 – OSSUPHx Output Selection Set for PWMH output of the channel x

Value	Description
0	No effect.
1	Output override value OOVHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

68.7.23 PWM Output Selection Clear Update Register

Name: PWM_OSCUPD
Offset: 0x58
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					OSCUPL3	OSCUPL2	OSCUPL1	OSCUPL0
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OSCUPH3	OSCUPH2	OSCUPH1	OSCUPH0
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19 – OSCUPLx Output Selection Clear for PWML output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

Bits 0, 1, 2, 3 – OSCUPHx Output Selection Clear for PWMH output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

68.7.24 PWM Fault Mode Register

Name: PWM_FMR
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

See Fault Inputs for details on fault generation.

⚠ CAUTION To prevent an unexpected activation of the status flag FSy in the PWM Fault Status Register, the bit FMODy can be set to '1' only if the FPOLy bit has previously configured to its final value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	FFIL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FMOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FPOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – FFIL[7:0] Fault Filtering
For each bit y of FFIL, where y is the fault input number:
0: The fault input y is not filtered.
1: The fault input y is filtered.

Bits 15:8 – FMOD[7:0] Fault Activation Mode
For each bit y of FMOD, where y is the fault input number:
0: The fault y is active until the fault condition is removed at the peripheral[(1)] level.
1: The fault y stays active until the fault condition is removed at the peripheral level[(1)]AND until it is cleared in the PWM Fault Clear Register.
Note:
1. The peripheral generating the fault.

Bits 7:0 – FPOL[7:0] Fault Polarity
For each bit y of FPOL, where y is the fault input number:
0: The fault y becomes active when the fault input y is at 0.
1: The fault y becomes active when the fault input y is at 1.

68.7.25 PWM Fault Status Register

Name: PWM_FSR
Offset: 0x60
Reset: 0x00000000
Property: Read-only

Refer to Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – FS[7:0] Fault Status
For each bit y of FS, where y is the fault input number:
0: The fault y is not currently active.
1: The fault y is currently active.

Bits 7:0 – FIV[7:0] Fault Input Value
For each bit y of FIV, where y is the fault input number:
0: The current sampled value of the fault input y is 0 (after filtering if enabled).
1: The current sampled value of the fault input y is 1 (after filtering if enabled).

68.7.26 PWM Fault Clear Register

Name: PWM_FCR
Offset: 0x64
Reset: –
Property: Write-only

See Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FCLR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – FCLR[7:0] Fault Clear
For each bit y of FCLR, where y is the fault input number:
0: No effect.
1: If bit y of FMOD field is set to '1' and if the fault input y is not at the level defined by the bit y of FPOL field, the fault y is cleared and becomes inactive (FMOD and FPOL fields belong to PWM Fault Mode Register), else writing this bit to '1' has no effect.

68.7.27 PWM Fault Protection Value Register 1

Name: PWM_FPV1
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

See Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FPVL3	FPVL2	FPVL1	FPVL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					FPVH3	FPVH2	FPVH1	FPVH0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19 – FPVLx Fault Protection Value for PWML output on channel x
This bit is taken into account only if the bit FPZLx is set to ‘0’ in PWM Fault Protection Value Register 2.

Value	Description
0	PWML output of channel x is forced to ‘0’ when fault occurs.
1	PWML output of channel x is forced to ‘1’ when fault occurs.

Bits 0, 1, 2, 3 – FPVHx Fault Protection Value for PWMH output on channel x
This bit is taken into account only if the bit FPZHx is set to ‘0’ in PWM Fault Protection Value Register 2.

Value	Description
0	PWMH output of channel x is forced to ‘0’ when fault occurs.
1	PWMH output of channel x is forced to ‘1’ when fault occurs.

68.7.28 PWM Fault Protection Enable Register

Name: PWM_FPE
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Only the first 6 bits (number of fault input pins) of fields FPEx are significant.

Refer to Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
	FPE3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FPE2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FPE1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FPE0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – FPEx Fault Protection Enable for channel x
For each bit y of FPEx, where y is the fault input number:
0: Fault y is not used for the fault protection of channel x.
1: Fault y is used for the fault protection of channel x.

⚠ **CAUTION** To prevent an unexpected activation of the fault protection, the bit y of FPEx field can be set to ‘1’ only if the corresponding FPOL field has been previously configured to its final value in PWM Fault Mode Register.

68.7.29 PWM Event Line x Mode Register

Name: PWM_ELMRx
Offset: 0x7C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CSELy Comparison y Selection

Value	Description
0	A pulse is not generated on the event line x when the comparison y matches.
1	A pulse is generated on the event line x when the comparison y match.

68.7.30 PWM Spread Spectrum Register

Name: PWM_SSPR
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
								SPRDM
Access								R/W
Reset								0
Bit	23	22	21	20	19	18	17	16
	SPRD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SPRD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SPRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – SPRDM Spread Spectrum Counter Mode

Value	Description
0	Triangular mode. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled and counts upwards at each PWM period. When it reaches +SPRD, it restarts to count from -SPRD again.
1	Random mode. The spread spectrum counter is loaded with a new random value at each PWM period. This random value is uniformly distributed and is between -SPRD and +SPRD.

Bits 23:0 – SPRD[23:0] Spread Spectrum Limit Value
The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying PWM period for the output waveform.

68.7.31 PWM Spread Spectrum Update Register

Name: PWM_SSPUP
Offset: 0xA4
Reset: –
Property: Write-only

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the SPRD value. This prevents an unexpected waveform when modifying the spread spectrum limit value.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SPRDUP[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	SPRDUP[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SPRDUP[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – SPRDUP[23:0] Spread Spectrum Limit Value Update
The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying period for the output waveform.

68.7.32 PWM Debug Register

Name: PWM_DEBUG
Offset: 0xAC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								OUTMODE
Access								R/W
Reset								0

Bit 0 – OUTMODE PWM Output Mode when System is in Debug Mode
0 (NO_EFFECT): Keeps the PWM outputs running when the processor reports a debug operating mode.
1 (STUCK_AT): Forces the PWM outputs with the values configured in PWM_FPVx as soon as the processor reports a debug operating mode. See PWM_FPV1 and PWM_FPV2.

68.7.33 PWM Stepper Motor Mode Register

Name: PWM_SMMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DOWN1	DOWN0
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							GCEN1	GCEN0
Access							R/W	R/W
Reset							0	0

Bits 16, 17 – DOWNx Down Count

Value	Description
0	Up counter.
1	Down counter.

Bits 0, 1 – GCENx Gray Count Enable

Value	Description
0	Disable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1]
1	Enable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1].

68.7.34 PWM Fault Protection Value Register 2

Name: PWM_FPV2
Offset: 0xC0
Reset: 0x000F000F
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
					FPZL3	FPZL2	FPZL1	FPZL0
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
					FPZH3	FPZH2	FPZH1	FPZH0
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 16, 17, 18, 19 – FPZLx Fault Protection to Hi-Z for PWML output on channel x

Value	Description
0	When fault occurs, PWML output of channel x is forced to value defined by the bit FPVLx in PWM Fault Protection Value Register 1.
1	When fault occurs, PWML output of channel x is forced to high-impedance state.

Bits 0, 1, 2, 3 – FPZHx Fault Protection to Hi-Z for PWMH output on channel x

Value	Description
0	When fault occurs, PWMH output of channel x is forced to value defined by the bit FPVHx in PWM Fault Protection Value Register 1.
1	When fault occurs, PWMH output of channel x is forced to high-impedance state.

68.7.35 PWM Write Protection Control Register

Name: PWM_WPCR
Offset: 0xE4
Reset: –
Property: Write-only

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD[1:0]	
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50574D	PASSWD	Writing any other value in this field aborts the write operation of the WPCMD field. Always reads as 0

Bits 2, 3, 4, 5, 6, 7 – WPRGx Write Protection Register Group x

Value	Description
0	The WPCMD command has no effect on the register group x.
1	The WPCMD command is applied to the register group x.

Bits 1:0 – WPCMD[1:0] Write Protection Command
This command is performed only if the WPKEY corresponds to 0x50574D (“PWM” in ASCII).

Value	Name	Description
0	DISABLE_SW_PROT	Disables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
1	ENABLE_SW_PROT	Enables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
2	ENABLE_HW_PROT	Enables the hardware write protection of the register groups of which the bit WPRGx is at ‘1’. Only a hardware reset of the PWM controller can disable the hardware write protection. Moreover, to meet security requirements, the PIO lines associated with the PWM can not be configured through the PIO interface.

68.7.36 PWM Write Protection Status Register

Name: PWM_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WPVS		WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bits 31:16 – WPVSRC[15:0] Write Protect Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bits 8, 9, 10, 11, 12, 13 – WPHWSx Write Protect HW Status

Value	Description
0	The HW write protection x of the register group x is disabled.
1	The HW write protection x of the register group x is enabled.

Bit 7 – WPVS Write Protect Violation Status

Value	Description
0	No write protection violation has occurred since the last read of PWM_WPSR.
1	At least one write protection violation has occurred since the last read of PWM_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Bits 0, 1, 2, 3, 4, 5 – WPSWSx Write Protect SW Status

Value	Description
0	The SW write protection x of the register group x is disabled.
1	The SW write protection x of the register group x is enabled.

68.7.37 PWM Comparison x Value Register

Name: PWM_CMPVx
Offset: 0x0130 + x*0x10 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Only the first 16 bits (channel counter size) of field CV are significant.

Bit	31	30	29	28	27	26	25	24
								CVM
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	CV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CVM Comparison x Value Mode

Value	Description
0	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.
1	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing. Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

Bits 23:0 – CV[23:0] Comparison x Value
Define the comparison x value to be compared with the counter of the channel 0.

68.7.38 PWM Comparison x Value Update Register

Name: PWM_CMPVUPDx
Offset: 0x0134 + x*0x10 [x=0..7]
Reset: –
Property: Write-only

This register acts as a double buffer for the CV and CVM values. This prevents an unexpected comparison x match.

Only the first 16 bits (channel counter size) of field CVUPD are significant.

⚠ CAUTION The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

Bit	31	30	29	28	27	26	25	24
								CVMUPD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	CVUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CVUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CVUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – CVMUPD Comparison x Value Mode Update
Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

Value	Description
0	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.
1	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing.

Bits 23:0 – CVUPD[23:0] Comparison x Value Update
Defines the comparison x value to be compared with the counter of the channel 0.

68.7.39 PWM Comparison x Mode Register

Name: PWM_CMPMx
Offset: 0x0138 + x*0x10 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CUPRCNT[3:0]				CUPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPRCNT[3:0]				CPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CTR[3:0]							CEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 23:20 – CUPRCNT[3:0] Comparison x Update Period Counter
Reports the value of the comparison x update period counter.
Note: The field CUPRCNT is read-only

Bits 19:16 – CUPR[3:0] Comparison x Update Period
Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

Bits 15:12 – CPRCNT[3:0] Comparison x Period Counter
Reports the value of the comparison x period counter.
Note: The field CPRCNT is read-only

Bits 11:8 – CPR[3:0] Comparison x Period
CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

Bits 7:4 – CTR[3:0] Comparison x Trigger
The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

Bit 0 – CEN Comparison x Enable

Value	Description
0	The comparison x is disabled and can not match.
1	The comparison x is enabled and can match.

68.7.40 PWM Comparison x Mode Update Register

Name: PWM_CMPMUPDx
Offset: 0x013C + x*0x10 [x=0..7]
Reset: –
Property: Write-only

This register acts as a double buffer for the CEN, CTR, CPR and CUPR values. This prevents an unexpected comparison x match.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CUPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
					CPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CTRUPD[3:0]							CENUPD
Access	W	W	W	W				W
Reset	–	–	–	–				–

Bits 19:16 – CUPRUPD[3:0] Comparison x Update Period Update
Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

Bits 11:8 – CPRUPD[3:0] Comparison x Period Update
CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

Bits 7:4 – CTRUPD[3:0] Comparison x Trigger Update
The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

Bit 0 – CENUPD Comparison x Enable Update

Value	Description
0	The comparison x is disabled and can not match.
1	The comparison x is enabled and can match.

68.7.41 PWM Channel Mode Register

Name: PWM_CMRx
Offset: 0x0200 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PPM	DTLI	DTHI	DTE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TCTS	DPOLI	UPDS	CES	CPOL	CALG
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CPRE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – PPM Push-Pull Mode
The Push-Pull mode is enabled for channel x.

Value	Description
0	The Push-Pull mode is disabled for channel x.
1	The Push-Pull mode is enabled for channel x.

Bit 18 – DTLI Dead-Time PWMLx Output Inverted

Value	Description
0	The dead-time PWMLx output is not inverted.
1	The dead-time PWMLx output is inverted.

Bit 17 – DTHI Dead-Time PWMHx Output Inverted

Value	Description
0	The dead-time PWMHx output is not inverted.
1	The dead-time PWMHx output is inverted.

Bit 16 – DTE Dead-Time Generator Enable

Value	Description
0	The dead-time generator is disabled.
1	The dead-time generator is enabled.

Bit 13 – TCTS Timer Counter Trigger Selection

Value	Description
0	The comparator of the channel x (OCx) is used as the trigger source for the Timer Counter (TC).
1	The counter events of the channel x is used as the trigger source for the Timer Counter (TC).

Bit 12 – DPOLI Disabled Polarity Inverted

Value	Description
0	When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is the same as the one defined by the CPOL bit.
1	When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is inverted compared to the one defined by the CPOL bit.

Bit 11 – UPDS Update Selection
If the PWM period is center-aligned (CALG=1):
0: The update occurs at the next end of the PWM period after writing the update register(s).
1: The update occurs at the next end of the PWM half period after writing the update register(s).
If the PWM period is left-aligned (CALG=0), the update always occurs at the end of the PWM period after writing the update register(s).

Bit 10 – CES Counter Event Selection
If the PWM period is center-aligned (CALG=1):
0: The channel counter event occurs at the end of the PWM period.
1: The channel counter event occurs at the end of the PWM period and at half the PWM period.
If the PWM period is left-aligned (CALG=0), the channel counter event occurs at the end of the period and the CES bit has no effect.

Bit 9 – CPOL Channel Polarity

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

Bit 8 – CALG Channel Alignment

Value	Description
0	The period is left-aligned.
1	The period is center-aligned.

Bits 3:0 – CPRE[3:0] Channel Prescaler

Value	Name	Description
	MCK	Peripheral clock
1	MCK_DIV_2	Peripheral clock/2
2	MCK_DIV_4	Peripheral clock/4
3	MCK_DIV_8	Peripheral clock/8
4	MCK_DIV_16	Peripheral clock/16
5	MCK_DIV_32	Peripheral clock/32
6	MCK_DIV_64	Peripheral clock/64
7	MCK_DIV_128	Peripheral clock/128
8	MCK_DIV_256	Peripheral clock/256
9	MCK_DIV_512	Peripheral clock/512
10	MCK_DIV_1024	Peripheral clock/1024
11	CLKA	Clock A
12	CLKB	Clock B

68.7.42 PWM Channel Duty Cycle Register

Name: PWM_CDTYx
Offset: 0x0204 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CDTY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CDTY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CDTY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CDTY[23:0] Channel Duty-Cycle
Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

68.7.43 PWM Channel Duty Cycle Update Register

Name: PWM_CDTYUPDx
Offset: 0x0208 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register acts as a double buffer for the CDTY value. This prevents an unexpected waveform when modifying the waveform duty-cycle.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CDTYUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CDTYUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CDTYUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – CDTYUPD[23:0] Channel Duty-Cycle Update
Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

68.7.44 PWM Channel Period Register

Name: PWM_CPRDx
Offset: 0x020C + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CPRD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CPRD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CPRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CPRD[23:0] Channel Period
If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value "X" (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(2 \times X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value "X" (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(2 \times X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(2 \times X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

68.7.45 PWM Channel Period Update Register

Name: PWM_CPRDUPDx
Offset: 0x0210 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPRD value. This prevents an unexpected waveform when modifying the waveform period.

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CPRDUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CPRDUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CPRDUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – CPRDUPD[23:0] Channel Period Update
If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times \mathrm{CPRDUPD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times \mathrm{CPRDUPD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(X \times \mathrm{CPRDUPD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(2 \times X \times \mathrm{CPRDUPD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$\frac{(2 \times X \times \text{CPRDUPD} \times \text{DIVA})}{f_{\text{peripheral clock}}}$ or $\frac{(2 \times X \times \text{CPRDUPD} \times \text{DIVB})}{f_{\text{peripheral clock}}}$

68.7.46 PWM Channel Counter Register

Name: PWM_CCNTx
Offset: 0x0214 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read-only

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CNT[23:0] Channel Counter Register
Channel counter value. This register is reset when:
- the channel is enabled (writing CHIDx in the PWM_ENA register).
- the channel counter reaches CPRD value defined in the PWM_CPRDx register if the waveform is left-aligned.

68.7.47 PWM Channel Dead Time Register

Name: PWM_DTx
Offset: 0x0218 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

Only the first 12 bits (dead-time counter size) of fields DTH and DTL are significant.

Bit	31	30	29	28	27	26	25	24
	DTL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DTL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DTH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DTL[15:0] Dead-Time Value for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx).

Bits 15:0 – DTH[15:0] Dead-Time Value for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx).

68.7.48 PWM Channel Dead Time Update Register

Name: PWM_DTUPDx
Offset: 0x021C + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the DTH and DTL values. This prevents an unexpected waveform when modifying the dead-time values.

Only the first 12 bits (dead-time counter size) of fields DTHUPD and DTLUPD are significant.

Bit	31	30	29	28	27	26	25	24
	DTLUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	DTLUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	DTHUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	DTHUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:16 – DTLUPD[15:0] Dead-Time Value Update for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

Bits 15:0 – DTHUPD[15:0] Dead-Time Value Update for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

68.7.49 PWM Channel Mode Update Register

Name: PWM_CMUPDx
Offset: 0x0400 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPOL value. This prevents an unexpected waveform when modifying the polarity value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CPOLINVUP				CPOLUP	
Access			W				W	
Reset			–				–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 13 – CPOLINVUP Channel Polarity Inversion Update
If this bit is written at ‘1’, the write of the bit CPOLUP is not taken into account.

Value	Description
0	No effect.
1	The OCx output waveform (output from the comparator) is inverted.

Bit 9 – CPOLUP Channel Polarity Update
The write of this bit is taken into account only if the bit CPOLINVUP is written at ‘0’ at the same time.

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

68.7.50 PWM External Trigger Register

Name: PWM_ETRGx
Offset: 0x042C + (x-1)*0x20 [x=1..2]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
	MAXCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MAXCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAXCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – RFEN Recoverable Fault Enable

Value	Description
0	The TRGINx signal does not generate a recoverable fault.
1	The TRGINx signal generate a recoverable fault in place of the fault x input.

Bit 30 – TRGSRC Trigger Source

Value	Description
0	The TRGINx signal is driven by the PWMEXTRGx input.
1	

Bit 29 – TRGFILT Filtered input

Value	Description
0	The external trigger input x is not filtered.
1	The external trigger input x is filtered.

Bit 28 – TRGEDGE Edge Selection

Value	Name	Description
0	FALLING_ZERO	TRGMODE = 1: TRGINx event detection on falling edge. TRGMODE = 2, 3: TRGINx active level is 0
1	RISING_ONE	TRGMODE = 1: TRGINx event detection on rising edge. TRGMODE = 2, 3: TRGINx active level is 1

Bits 25:24 – TRGMODE[1:0] External Trigger Mode

Value	Name	Description
0	OFF	External trigger is not enabled.
1	MODE1	External PWM Reset Mode
2	MODE2	External PWM Start Mode
3	MODE3	Cycle-by-cycle Duty Mode

Bits 23:0 – MAXCNT[23:0] Maximum Counter value
Maximum channel x counter value measured at the TRGINx event since the last read of the register.
At the TRGINx event, if the channel x counter value is greater than the stored MAXCNT value, then MAXCNT is updated by the channel x counter value.

68.7.51 PWM Leading-Edge Blanking Register

Name: PWM_LEBRx
Offset: 0x0430 + (x-1)*0x20 [x=1..2]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LEBDELAY[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 19 – PWMHREN PWMH Rising Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMHx output rising edge.
1	Leading-edge blanking is enabled on PWMHx output rising edge.

Bit 18 – PWMHFEN PWMH Falling Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMHx output falling edge.
1	Leading-edge blanking is enabled on PWMHx output falling edge.

Bit 17 – PWMLREN PWML Rising Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMLx output rising edge.
1	Leading-edge blanking is enabled on PWMLx output rising edge.

Bit 16 – PWMLFEN PWML Falling Edge Enable

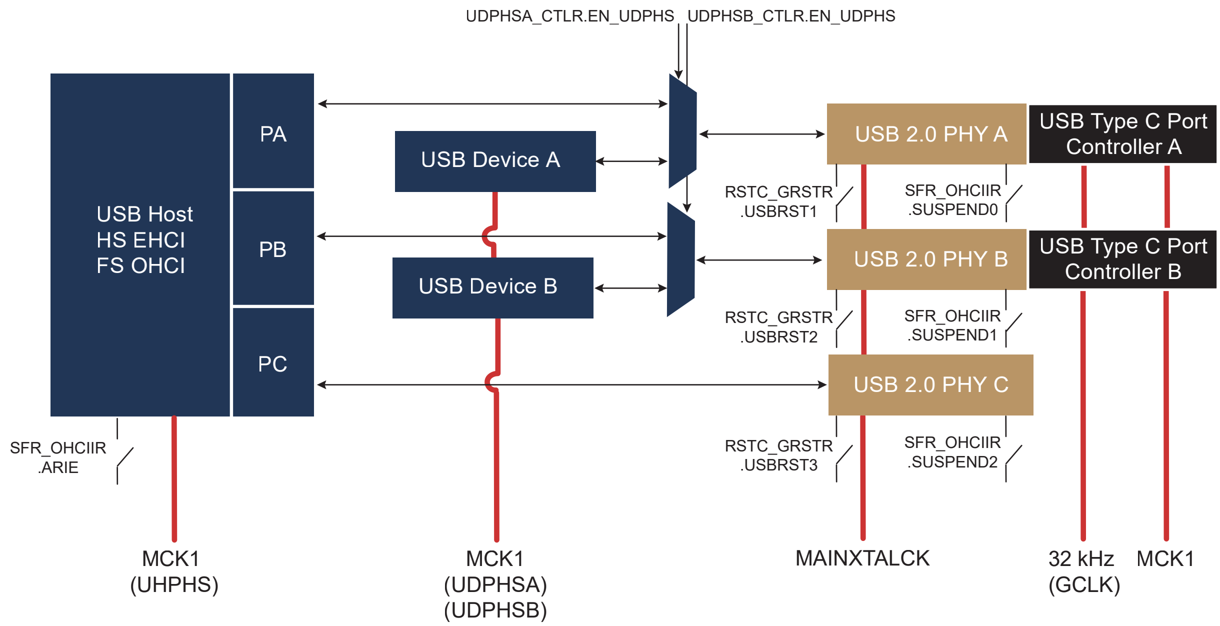
Value	Description
0	Leading-edge blanking is disabled on PWMLx output falling edge.
1	Leading-edge blanking is enabled on PWMLx output falling edge.

Bits 6:0 – LEBDELAY[6:0] Leading-Edge Blanking Delay for TRGINx
Leading-edge blanking duration for external trigger x input. The delay is calculated according to the following formula:
LEBDELAY = ($f_{\text{peripheral clock}}$ × Delay) + 1

69. USB SUBSYSTEM

69.1 Block Diagram

Figure 69-1. USB Subsystem Block Diagram



69.2 Components

- 1x high-speed host (EHCI) and one full-speed host (OHCI) with three ports (PA, PB and PC) (UHPHS)
- 2x high-speed devices (Device A and Device B) shared with PA and PB (UDPHS)
- 3x USB 2.0 PHYs
- 2x USB Type-C Port Controller (TCPC)

69.3 Product Dependencies

69.3.1 Clocks

The device features three USB 2.0 PHYs and each one has its own PLL. Each PLL is fed with the main crystal oscillator. The PLL starts as soon as PHY reset is released in RSTC_GRSTR.USB_RSTx. There is no control in the Power Management Controller (PMC). Refer to the section Electrical Characteristics.

The TCPC is clocked through the PMC. First configure the PMC to enable the TCPC peripheral clock (MCK1) and the TCPC 32 kHz generic clock (GCLK).

UHP12M, UHP48M and OHCI clocks are generated from Transceiver A so, for OHCI operations, Port A must be enabled.

If the USB port is suspended, the clocks to the UTMI of that port are stopped. As Port A UTMI controls the UTMI clocks for Ports B and C, stopping the UTMI clocks on Port A also prevents Ports B and C UTMIs from being clocked.

Clearing the SFR_UTMI0Rx.COMMONONN bit for each port prevents the USB suspend from stopping the UTMI clocks for that port.

To clear COMMONONN:

1. Reset the USB port (by setting RSTC_GRSTR.USB_RSTx).
2. Clear the COMMONONN bit.
3. Release the USB port reset (by clearing USB_RSTx).

Note: At product startup, if no bootable code is available to the product, by default Port A is configured in USB Device mode and suspended.

To enable access to Ports B and C programming interfaces, take Port A out of the USB Suspend state, either by:

- disabling the USB Device mode on Port A by clearing UDPHS_CTRL.EN_UDPHS, or by
- clearing the COMMONONN bit as described above.

69.3.2 Interrupts

Refer to the table Peripheral Identifiers.

69.3.3 Reset

USB peripherals are connected to the processor and peripherals reset line.

USB 2.0 PHY resets are controlled with the RSTC_GRSTR.USB_RSTx bits.

⚠ **CAUTION** To use Type C port controller, the corresponding USB 2.0 PHY reset line must be released.

69.3.4 I/Os

USB I/Os are dedicated I/Os powered by specific rail VDDUTMII/GNDUTMI.

69.4 Special Functions in SFR/SFRBU

- SFR_OHCIICR.SUSPENDx to control suspend for each port. These bits must be set to reduce power consumption on VDDUTMII in Low-power mode.
- SFR_OHCIICR.ARIE to enable OHCI asynchronous resume interrupt
- SFR_OHCIISR.RISx for Port Resume status of each port
- SFR_UTMI0R, SFR_UTMI1R, SFR_UTMI2R for UTMI configuration, VBUS status and power control, in particular SFR_UTMI0Rx.COMMONONN

70. USB Type-C™ Port Controller (TCPC)

70.1 Reference Documents

Table 70-1. Reference Documents

Owner - Reference	Title
USB Implementers Forum, Inc.	Battery Charging Specification Revision 1.2 December 7, 2010
USB 3.0 Promoter Group	Universal Serial Bus Type-C™ Port Controller Interface Specification Revision 1.0, Version 1.2 November 2016
USB 3.0 Promoter Group	Universal Serial Bus Type-C Cable and Connector Specification Revision 1.2 March 25, 2016

70.2 Description

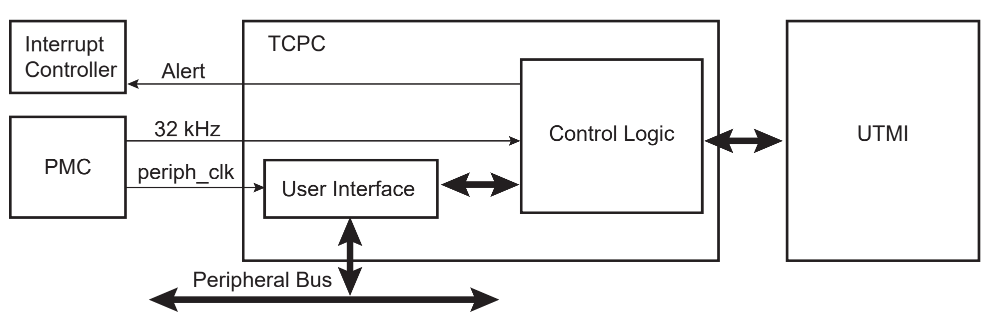
The USB Type-C™ Port Controller (TCPC) provides a low level interface to the USB Battery Charging and USB Type-C PHY. It also features a register set compliant with the Universal Serial Bus Type-C™ Port Controller Interface Specification with an access to these registers that enables a software implementation of the Universal Serial Bus Type-C™ Port Controller Interface Specification. See Reference Documents.

70.3 Embedded Characteristics

- Battery Charging Source Type Detection
- Type-C Cable Connection Detection
- Type-C Power Source Advertising 5V, 500 mA/1.5A/3.0A
- Type-C Power Capability Detection 5V, 500 mA/1.5A/3.0A
- Enables USB.org Compliant Software Implementation of USB2.0 TCPC

70.4 Block Diagram

Figure 70-1. TCPC Block Diagram



70.5 Product Dependencies

70.5.1 Power Management

The TCPC may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the TCPC peripheral clock (periph_clk) and the TCPC 32 kHz generic clock (GCLK).

70.5.2 Interrupt Sources

The TCPC has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TCPC.

70.6 Functional Description

The TCPC provides access to the USB Phy for the software implementation of the Universal Serial Bus Type-C™ Port Controller Interface Specification and the Battery Charging Specification.

70.6.1 Battery Charging

The Battery Charging Specification defines the charger detection hardware to be controlled.

The features defined in the Battery Charging Specification can be controlled with the TCPC_UPC register and the resulting outputs can be observed in the TCPC_UPS as shown in the table and figure that follow.

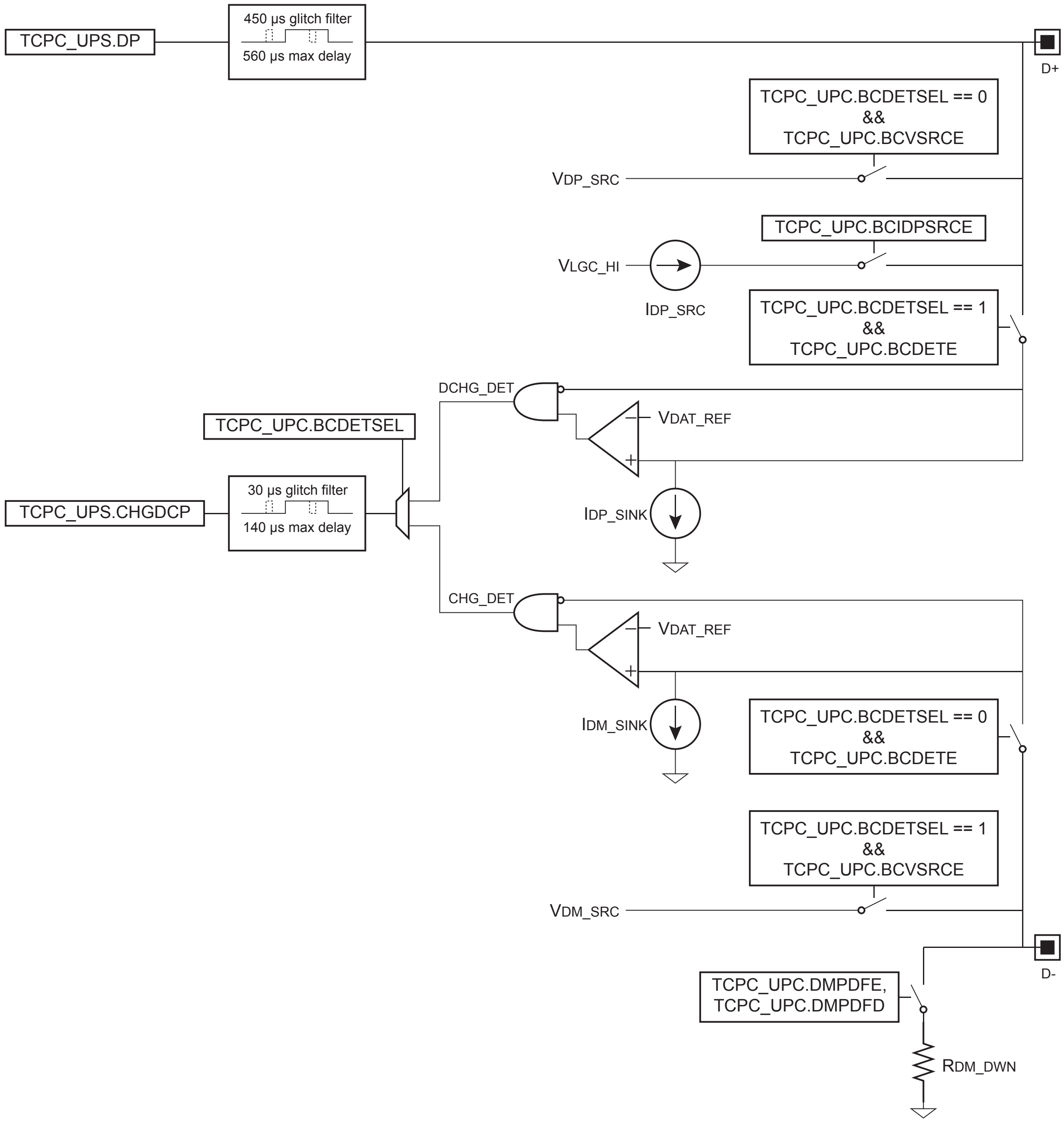
Table 70-2. Battery Charging Specification to USB Phy Mapping

Battery Charging Specification Reference	TCPC Register Field
Data Contact Detect I_{DP_SRC} Current Source Enable	TCPC_UPC.BCIDPSRCE
Data Contact Detect R_{DM_DWN} Pull Down Control	Force activation: TCPC_UPC.DMPDFE Force deactivation: TCPC_UPC.DMPDFD
Primary Detection or Secondary Detection	Select detection type: TCPC_UPC.BCDETSEL
Primary / Secondary Detection V_{DP_SRC} / V_{DM_SRC} Voltage Source Enable	TCPC_UPC.BCVSRCE
Primary / Secondary Detection CHG_DET / DCHG_DET(1) Comparator Enable	TCPC_UPC.BCDETE
Primary / Secondary Detection CHG_DET / DCHG_DET(1) Comparator Output	TCPC_UPS.CHGDCP
Data Contact Detect D+ line state	TCPC_UPS.DP

Note:

1. In the Battery Charging Specification, DCHG_DET indicates DCP_DET comparator output in a Portable Device (PD) and indicates PRTBL_DET comparator output in a Charging Downstream Port (CDP).

Figure 70-2. Charger Detection Hardware



After any software change in the USB Phy Control register (TCPC_UPC) or after any change on the USB D+ line state, the charger detection outputs in the USB Phy Status register (TCPC_UPS) may not be valid before an update delay, as shown in the Max Delay Update column of the table below.

These update delays must be respected after any TCPC_UPC software change when implementing the Data Contact Detect and Primary / Secondary Detection of the "Charger Detection Algorithms" specified in the Battery Charging Specification.

Changes to TCPC_UPS register bits are filtered to avoid USB D+ line state glitches propagation. The maximum glitch duration that is always filtered is shown in the Min Glitch Filtering column in the table below.

Any longer filtering or debouncing, such as during the t_{DCD_DBNC} time for the D+ line as specified in the Battery Charging Specification (see Reference Documents), must be performed by periodically sampling TCPC_UPS.

Table 70-3. TCPC_UPS Glitch Filtering Times

TCPC_UPS Bit	Max Delay Update	Min Glitch Filtering
TCPC_UPS.DP	560 µs	450 µs
TCPC_UPS.CHGDCP	140 µs	30 µs

70.6.2 USB Type-C

The TCPC enables a software implementation of the USB Type-C CC Control and sensing and of the USB Type-C Power Control for V_{BUS} as defined in the Universal Serial Bus Type-C™ Port Controller Interface Specification. The TCPC provides both a user interface compatible with the programmer model of the USB Type-C Port Controller and a direct access to the Configuration Channel (CC) hardware. V_{BUS} detection and control hardware is available through the PIO controller externally to the TCPC.

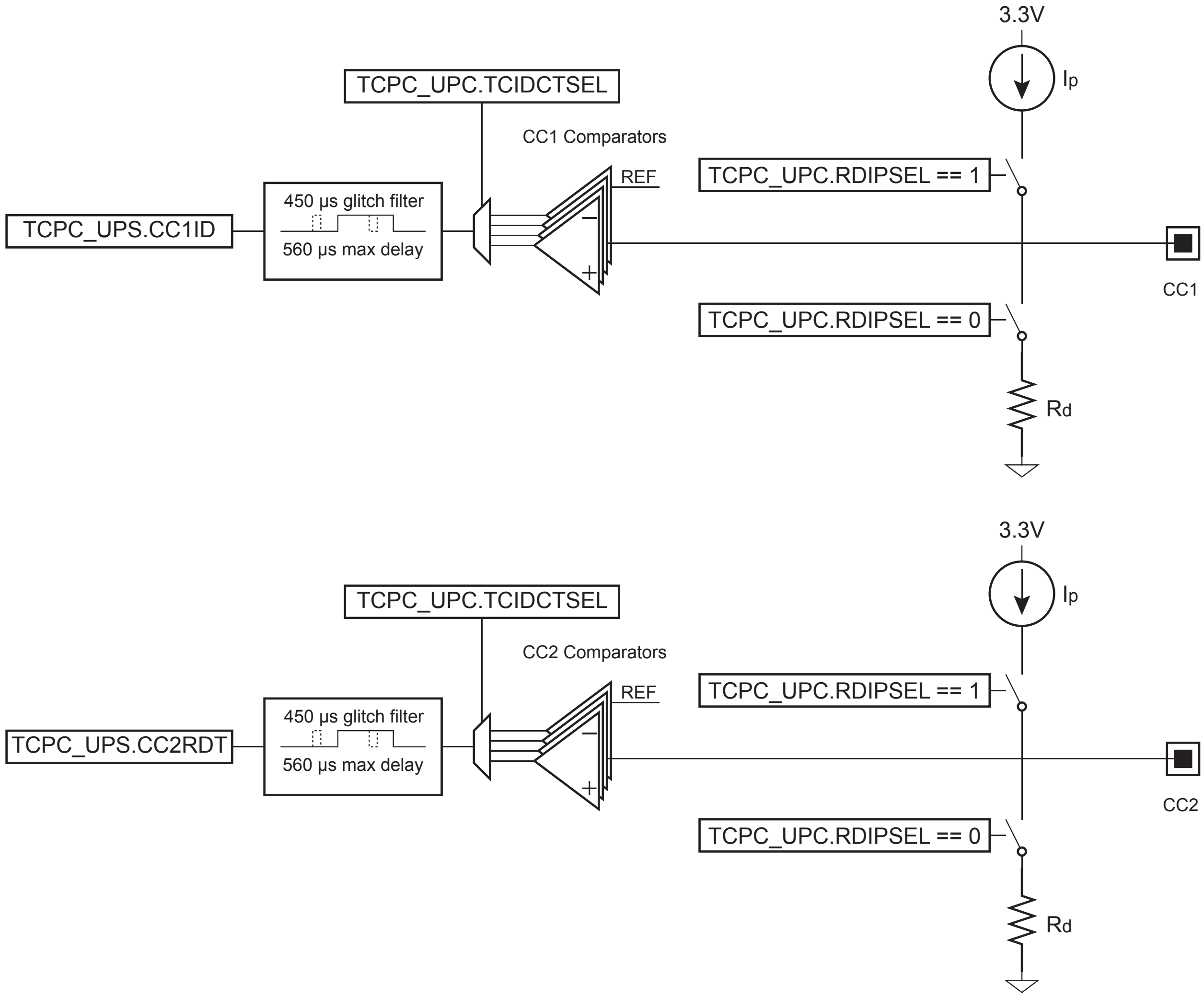
The Configuration Channel (CC) used in the discovery and configuration of connections across a USB Type-C cable is further defined in the Universal Serial Bus Type-C™ Port Controller Interface Specification.

The Configuration Channel (CC) hardware can be controlled by TCPC_UPC, and the resulting outputs can be observed in TCPC_UPS, as shown in the table and figure that follow.

Table 70-4. Configuration Channel (CC) Hardware to USB Phy Mapping

Configuration Channel Hardware	TCPC Register Field
CC Pull Up Rp	TCPC_UPC.RDIPSEL
CC Pull Down Rd	TCPC_UPC.RDIPSEL
CC Status Comparator Threshold	TCPC_UPC.TCIDCTSEL
CC Status Comparator Outputs	TCPC_UPS.CC2RDT

Figure 70-3. Configuration Channel (CC) Hardware



After any software change in TCPC_UPC or after any change on the CC1 or CC2 pins, the CC comparator outputs in TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT are not valid before some update delay as shown in the Max Delay Update column of the table below.

These update delays must be respected after any TCPC_UPC software change when implementing the CC control and sensing as specified by the Universal Serial Bus Type-C™ Port Controller Interface Specification.

The CC comparator outputs in TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT are debounced by a glitch filter rejecting any pulse of $t_{TCPCfilter}$ = 450 µs or less as shown in the Min Glitch Filtering column of the table below.

Any longer filtering or debouncing, such as during the $t_{CCDebounce}$ time as specified in the Universal Serial Bus Type-C Cable and Connector Specification, must be performed by periodically sampling TCPC_UPS.

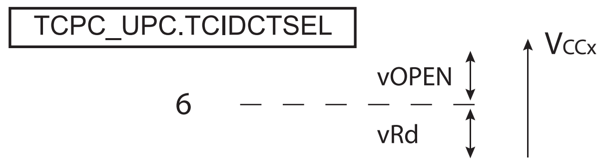
Table 70-5. TCPC_UPS Debouncing Times

TCPC_UPS Bit	Max Delay Update	Min Glitch Filtering
TCPC_UPS.CC1ID	560 µs	450 µs
TCPC_UPS.CC2RDT	560 µs	450 µs

70.6.2.1 Sink Detection at Power Source

Each of the three possible values—Rd or zOPEN—for the pull down resistor on CC1 and CC2 of the sink specified in the Universal Serial Bus Type-C Cable and Connector Specification (see Reference Documents) corresponds to one of the respective VRd or VOPEN voltage detection values at the source as shown in the figure below.

Figure 70-4. TCPC_UPC.TCIDCTSEL Values of CCx Voltage Thresholds at Source



For the purpose of connect detection at the source, the CC1 and CC2 voltage values can be determined by the following example procedure:

1. Set TCPC_UPC.RDIPSEL to the appropriate power advertising a non-zero value. Set TCPC_UPC.TCIDCTSEL to 6.
2. Wait for 560 µs or more.
   a. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT comparator outputs are equal to 00b indicating VOPEN voltage value on the CC pins, loop on step 2.
3. Wait for 450 µs.
   a. If TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT are equal to the previous sampled values, loop on step 3 until $t_{CCDebounce}$ time has elapsed.

   b. If TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT have changed, go back to step 2.
4. If TCPC_UPS.CC1ID or TCPC_UPS.CC2RDT is 1, the corresponding CCx voltage is VRd, else it is VOPEN.

Note: Only one TCPC_UPC write is required at step 1.

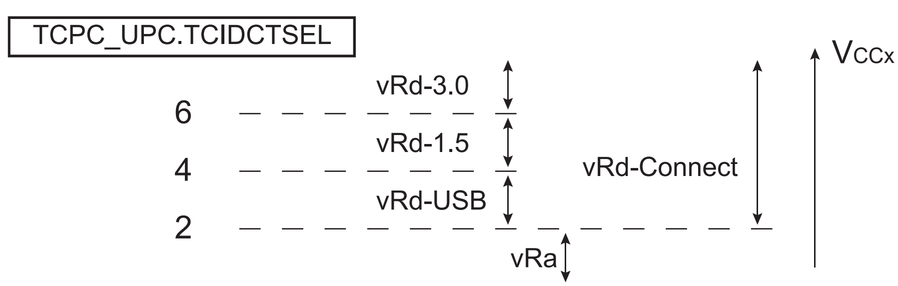
Any other sequence to determine the CC1ID and CC2RDT voltage values is permitted at any time as long as TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT delay update is respected.

If interfacing to a Type-C Port Manager (TCPM), the software has to update the CC Status register (TCPC_CCS) using the Set Status (TCPC_SSR) and Clear Status (TCPC_CSR) registers.

70.6.2.2 Power Source Detection at Sink

As defined in the in the Universal Serial Bus Type-C Cable and Connector Specification (see Reference Documents) , four voltage ranges may be present on CC1 and CC2 when connecting a Type-C cable. These voltage ranges are shown in the figure below.

Figure 70-5. TCPC_UPC.TCIDCTSEL Values of CCx Voltage Thresholds at Sink



Once V_{BUS} has been detected with the PIO controller, the sink may want to know the power delivery capability of the source. For that purpose the CC1 and CC2 voltage values can be determined by the following example procedure:

1. Set TCPC_UPC.RDIPSEL to 0 and TCPC_UPC.TCIDCTSEL to 2.
2. Wait for 560 µs or more.
   a. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT comparator outputs are equal to 00b indicating VRa voltage value on the CC pins, loop on step 2.
3. Wait for 450 µs.

a. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT are equal to the previous sampled values, loop on step 3 until $t_{CCDebounce}$ time has elapsed.

b. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT have changed, go back to step 2.

4. set TCPC_UPC.TCIDCTSEL to 4
5. Wait for 560 µs.
   a. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT that was 1 at step 3 is now 0, the CCx voltage is VRd-USB. Exit the sequence.

   b. Otherwise, set TCPC_UPC.TCIDCTSEL to 6 and go to step 6.
6. Wait for 560 µs.
   a. If the TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT that was 1 at step 6 is now 0, the CCx voltage is VRd-1.5, else it is VRd-3.0.

   Note: Only one TCPC_UPC write is required at each of steps 1, 4 and 5.

Any other sequence to determine the CC1ID and CC2RDT voltage values is permitted at any time as long as TCPC_UPS.CC1ID and TCPC_UPS.CC2RDT delay update is respected.

If interfacing to a TCPM, the software has to update TCPC_CCS using TCPC_SSR and TCPC_CSR.

70.6.2.3 USB-Type C Port Controller Status Register Management

The TCPC software is intended to regularly poll the TCPC_UPS register to meet the Universal Serial Bus Type-C™ Port Controller Interface Specification CC debouncing requirements, in particular after V_{BUS} detection or driving.

TCPC_SSR and TCPC_CSR are then used for the software to change the state of the CC Status (TCPC_CCS), Power Status (TCPC_PS), Fault Status (TCPC_FS) and Alert (TCPC_AL) registers.

The TCPC Role Control (TCPC_RCTL), Fault Control (TCPC_FCTL) and Command (TCPC_CMD) registers are to be read by the software to control the USB-Type C hardware through TCPC_UPC and external controls such as V_{BUS}.

The bits in registers TCPC_CCS, TCPC_PS and TCPC_FS are set from the Set Status (TCPC_SSR) register.

The bits in registers TCPC_CCS and TCPC_PS are cleared from the Clear Status (TCPC_CSR) register.

TCPC_AL is set from the Set Alert (TCPC_SAR) register.

The bits in TCPC_FS and TCPC_AL are cleared by writing each bit in these registers to ‘1’. Refer to the Universal Serial Bus Type-C™ Port Controller Interface Specification (see Reference Documents).

Note that any unmasked change in TCPC_FS or TCPC_PS, or any change in TCPC_CCS also automatically sets, respectively, FLT, PWRS and CCS bits in TCPC_AL.

Setting an unmasked bit in TCPC_AL triggers the Alert interrupt to the Interrupt Controller to signal status changes to the TCPM software.

Refer to the Universal Serial Bus Type-C™ Port Controller Interface Specification and Universal Serial Bus Type-C Cable and Connector Specification (see Reference Documents) for a detailed description of the algorithms to be implemented for supporting the various USB Type-C features.

70.7 Register Summary

Notes:

1. Refer to the Universal Serial Bus Type-C™ Port Controller Interface Specification (see Reference Documents) for a detailed description of the registers from offsets 0x00 to 0x29.
2. 8-bit or 16-bit subaccess to the TCPC registers is permitted within 32-bit address boundaries.
3. Simultaneous multiple register access with a single 32-bit or 16-bit transfer to contiguous 16-bit or 8-bit TCPC registers is permitted within 32-bit address boundaries.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TCPC_VID	15:8	VID[15:8]							
		7:0	VID[7:0]							
0x02	TCPC_PID	15:8	PID[15:8]							
		7:0	PID[7:0]							
0x04	TCPC_DID	15:8	DID[15:8]							
		7:0	DID[7:0]							
0x06	TCPC_UTCR	15:8	UTCR[15:8]							
		7:0	UTCR[7:0]							
0x08	TCPC_UPDR	15:8	UPDR[15:8]							
		7:0	UPDR[7:0]							
0x0A	TCPC_PDIR	15:8	PDIR[15:8]							
		7:0	PDIR[7:0]							
0x0C ... 0x0F	Reserved									
0x10	TCPC_AL	15:8	VDA				VBUSSNKDS		FLT	VBUSLO
		7:0	VBUSHI						PWRS	CCS
0x12	TCPC_ALM	15:8	VDA				VBUSSNKDS		FLT	VBUSLO
		7:0	VBUSHI						PWRS	CCS
0x14	TCPC_PSM	7:0		INIT		SRCVBUS	VBUSDETE	VBUS		SNKVBUS
0x15	TCPC_FSM	7:0	ALLREGRES	FRCOFVBUS	AUTDCHF	FRCDCHF	VBUSOCPF	VBUSOVPF		
0x16 ... 0x17	Reserved									
0x18	TCPC_CSO	7:0							CNX	COR
0x19	Reserved									
0x1A	TCPC_RCTL	7:0		DRP	RP[1:0]		CC2[1:0]		CC1[1:0]	
0x1B	TCPC_FCTL	7:0				FRCOFVBUS		VBUSOCPF	VBUSOVPF	
0x1C	Reserved									
0x1D	TCPC_CCS	7:0			LK4CNX	CONRES	CC2[1:0]		CC1[1:0]	
0x1E	TCPC_PS	7:0	DBG	INIT	SRCHIV	SRCVBUS	VBUSDETE	VBUS	VCONN	SNKVBUS
0x1F	TCPC_FS	7:0	ALLREGRES	FRCOFVBUS			VBUSOCPF	VBUSOVPF		
0x20 ... 0x22	Reserved									
0x23	TCPC_CMD	7:0	COMMAND[7:0]							
0x24	TCPC_DCP1	15:8		VBUSOCPR	VBUSOVPR	BLDDCH	FRCDCH	VBUSMSRAL	SRCRES[1:0]	
		7:0	ROLES[2:0]			SOPDBG	SRCVCN	SNKVBUS	SRCHVBUS	SRCVBUS
0x26	TCPC_DCP2	15:8								WDTMR
		7:0	SKDSCDET	STPDSCHTH	VBUSVALSB[1:0]		VCPSP[2:0]			VCOFC
0x28	TCPC_SIC	7:0						VBUSEOVF	VBUSEOCF	FOFVBUS
0x29	TCPC_SOC	7:0		DBGAI	VBUSPM	AAAI	ACI	MXCFGCTL	CNXPR	CNROR
0x2A ... 0x7F	Reserved									
0x80	TCPC_CR	31:24	WAKEY[23:16]							
		23:16	WAKEY[15:8]							
		15:8	WAKEY[7:0]							
		7:0								SWRST

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x84	TCPC_SSR	31:24	FAULT_STATUS[7:0]							
		23:16	POWER_STATUS[7:0]							
		15:8	CC_STATUS[7:0]							
		7:0								
0x88	TCPC_CSR	31:24								
		23:16	POWER_STATUS[7:0]							
		15:8	CC_STATUS[7:0]							
		7:0								
0x8C	TCPC_SAR	31:24								
		23:16								
		15:8	ALERT[15:8]							
		7:0	ALERT[7:0]							
0x90 ... 0x9F	Reserved									
0xA0	TCPC_UPC	31:24			BCDETE	BCVSRCE	BCDETSEL	BCIDPSRCE	DMPDFE	DMPDFD
		23:16								
		15:8			RDIPSEL[1:0]			TCIDCTSEL[2:0]		
		7:0								
0xA4	TCPC_UPS	31:24								
		23:16								
		15:8								
		7:0				CC2RDT	CC1ID	CHGDCP	DM	DP

70.7.1 TCPC Vendor ID Register

Name: TCPC_VID
Offset: 0x00
Reset: 0xBBBB
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	VID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	1	0	1	1

Bit	7	6	5	4	3	2	1	0
	VID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	1	0	1	1

Bits 15:0 – VID[15:0] Vendor ID
A unique 16-bit unsigned integer assigned by the USB-IF.

70.7.2 TCPC Product ID Register

Name: TCPC_PID
Offset: 0x02
Reset: 0xCCCC
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	PID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	0	1	1	0	0

Bit	7	6	5	4	3	2	1	0
	PID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	0	1	1	0	0

Bits 15:0 – PID[15:0] Product ID
A unique 16-bit unsigned integer assigned uniquely by the vendor to identify the TCPC.

70.7.3 TCPC Device ID Register

Name: TCPC_DID
Offset: 0x04
Reset: 0xDDDD
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	DID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	1	1	0	1

Bit	7	6	5	4	3	2	1	0
	DID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	1	1	0	1

Bits 15:0 – DID[15:0] Device ID
A unique 16-bit unsigned integer assigned by the vendor to identify the version of the TCPC.

70.7.4 TCPC USB Type-C Rev Register

Name: TCPC_UTCR
Offset: 0x06
Reset: 0xAAAA
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	UTCR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	UTCR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	0	1	0	1	0

Bits 15:0 – UTCR[15:0] USB Type-C Revision
Revision number assigned by USB-IF.

70.7.5 TCPC USB PD Rev Ver Register

Name: TCPC_UPDR
Offset: 0x08
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	UPDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UPDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – UPDR[15:0] USB-PD Specification Revision and Version
Always read as 0.

70.7.6 TCPC PD Interface Rev Register

Name: TCPC_PDIR
Offset: 0x0A
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	PDIR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PDIR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PDIR[15:0] USB-Port Controller Interface Specification Revision
Always read as 0.

70.7.7 TCPC Alert Register

Name: TCPC_AL
Offset: 0x10
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
	VDA				VBUSSNKDS		FLT	VBUSLO
Access	R/W				R/W		R/W	R/W
Reset	0				0		0	0

Bit	7	6	5	4	3	2	1	0
	VBUSHI						PWRS	CCS
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 15 – VDA Vendor Defined Alert
This bit can be cleared, regardless of the current status of the alert source.

Value	Description
0	No vendor defined alert has been detected.
1	A vendor defined alert has been detected.

Bit 11 – VBUSSNKDS VBUS Sink Disconnect Detected

Value	Description
0	No V_{BUS} sink disconnect threshold crossing has been detected.
1	A V_{BUS} sink disconnect threshold crossing has been detected.

Bit 9 – FLT Fault

Value	Description
0	No fault has occurred.
1	A fault has occurred. Read TCPC_FS.

Bit 8 – VBUSLO VBUS Voltage Alarm Low

Value	Description
0	A low-voltage alarm has not occurred.
1	A low-voltage alarm has occurred.

Bit 7 – VBUSHI VBUS Voltage Alarm High

Value	Description
0	A high-voltage alarm has not occurred.
1	A high-voltage alarm has occurred.

Bit 1 – PWRS Power Status

Value	Description
0	Power status not changed.
1	Power status changed.

Bit 0 – CCS CC Status

Value	Description
0	CC status not changed.
1	CC status changed.

70.7.8 TCPC Alert Mask Register

Name: TCPC_ALM
Offset: 0x12
Reset: 0x0FFF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Interrupt masked.

1: Interrupt unmasked.

Bit	15	14	13	12	11	10	9	8
	VDA				VBUSSNKDS		FLT	VBUSLO
Access	R/W				R/W		R/W	R/W
Reset	0				1		1	1

Bit	7	6	5	4	3	2	1	0
	VBUSHI						PWRS	CCS
Access	R/W						R/W	R/W
Reset	1						1	1

Bit 15 – VDA Vendor Defined Alert Interrupt Mask

Bit 11 – VBUSSNKDS VBUS Sink Disconnect Detected Interrupt Mask

Bit 9 – FLT Fault Interrupt Mask

Bit 8 – VBUSLO VBUS Voltage Alarm Low Interrupt Mask

Bit 7 – VBUSHI VBUS Voltage Alarm High Interrupt Mask

Bit 1 – PWRS Power Status Interrupt Mask

Bit 0 – CCS CC Status Interrupt Mask

70.7.9 TCPC Power Status Mask Register

Name: TCPC_PSM
Offset: 0x14
Reset: 0xFF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Interrupt masked.

1: Interrupt unmasked.

Bit	7	6	5	4	3	2	1	0
		INIT		SRCVBUS	VBUSDETE	VBUS		SNKVBUS
Access		R/W		R/W	R/W	R/W		R/W
Reset		1		1	1	1		1

Bit 6 – INIT TCPC Initialization Interrupt Mask

Bit 4 – SRCVBUS Sourcing VBUS Interrupt Mask

Bit 3 – VBUSDETE VBUS Present Detection Interrupt Mask

Bit 2 – VBUS VBUS Present Interrupt Mask

Bit 0 – SNKVBUS Sinking VBUS Interrupt Mask

70.7.10 TCPC Fault Status Mask Register

Name: TCPC_FSM
Offset: 0x15
Reset: 0xFF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Interrupt masked.

1: Interrupt unmasked.

Bit	7	6	5	4	3	2	1	0
	ALLREGRES	FRCOFVBUS	AUTDCHF	FRCDCHF	VBUSOCPF	VBUSOVPF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	1	1	1	1	1	1		

Bit 7 – ALLREGRES All Registers Reset To Default Interrupt Mask

Bit 6 – FRCOFVBUS Force Off VBUS Interrupt Mask

Bit 5 – AUTDCHF Auto Discharge Failed Interrupt Mask

Bit 4 – FRCDCHF Force Discharge Failed Interrupt Mask

Bit 3 – VBUSOCPF Internal or External OCP VBUS Over Current Protection Fault Interrupt Mask

Bit 2 – VBUSOVPF Internal or External OVP VBUS Over Voltage Protection Fault Interrupt Mask

70.7.11 TCPC Config Standard Output Register

Name: TCPC_CSO
Offset: 0x18
Reset: 0x60
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
							CNX	COR
Access							R/W	R/W
Reset							0	0

Bit 1 – CNX Connection Present

Value	Description
0	No connection (default).
1	Connection.

Bit 0 – COR Connector Orientation

Value	Description
0	Normal (CC1=A5, CC2=B5, TX1=A2/A3, RX1=B10/B11) default.
1	Flipped (CC2=A5, CC1=B5, TX1=B2/B3, RX1=A10/A11).

70.7.12 TCPC Role Control Register

Name: TCPC_RCTL
Offset: 0x1A
Reset: 0x0F
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
		DRP	RP[1:0]		CC2[1:0]		CC1[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	1	1	1	1

Bit 6 – DRP Dual Role Play
The TCPC must use the Rp value defined in the field RP when a connection is resolved, i.e., upon entry to Potential_Connect_as_Src in Figure 4-11. TCPC State Diagram of Universal Serial Bus Type-C™ Port Controller Interface Specification, before a connection
The TCPC toggles CC1 and CC2 after receiving TCPC_CMD.Look4Connection and until a connection is detected. Upon connection, the TCPC must resolve to either an Rp or Rd and report the CC1/CC2 State in TCPC_CCS. The CC pins must stay in Potential_Connect_as_Src or Potential_Connect_as_Sink until directed otherwise.

Value	Description
0	No DRP. CC1, CC2 fields determine Rp/Rd/Ra or open settings.
1	DRP.

Bits 5:4 – RP[1:0] Resistor for Power Advertising

Value	Name	Description
0	RP_DEFAULT	Rp default
1	RP_1P5A	Rp 1.5A
2	RP_3A	Rp 3.0A
3	RP_RESERVED	Reserved

Bits 3:2 – CC2[1:0] Configuration Channel 2

Value	Name	Description
0	CC2_RA	Ra
1	CC2_RP	Rp defined as in RP field
2	CC2_RD	Rd
3	CC2_OPEN	Open (Disconnect or don't care)

Bits 1:0 – CC1[1:0] Configuration Channel 1

Value	Name	Description
0	CC1_RA	Ra
1	CC1_RP	Rp defined as in RP field
2	CC1_RD	Rd
3	CC1_OPEN	Open (Disconnect or don't care)

70.7.13 TCPC Fault Control Register

Name: TCPC_FCTL
Offset: 0x1B
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
				FRCOFVBUS		VBUSOCPF	VBUSOVPF	
Access				R/W		R/W	R/W	
Reset				0		0	0	

Bit 4 – FRCOFVBUS Force Off VBUS (Source or Sink)
This enables or disables the standard input signal Force Off V_{BUS} functionality for debug purposes.

Value	Description
0	Allows standard input signal Force Off V_{BUS} control (default).
1	Blocks standard input signal Force Off V_{BUS} control. .

Bit 2 – VBUSOCPF Internal or External OCP VBUS Over Current Protection Fault

Value	Description
0	Internal and external OCP circuit enabled.
1	Internal and external OCP circuit disabled.

Bit 1 – VBUSOVPF Internal or External OVP VBUS Over Voltage Protection Fault

Value	Description
0	Internal and external OVP circuit enabled.
1	Internal and external OVP circuit disabled.

70.7.14 TCPC CC Status Register

Name: TCPC_CCS
Offset: 0x1D
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
			LK4CNX	CONRES	CC2[1:0]		CC1[1:0]	
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – LK4CNX Looking4Connection

Value	Description
0	TCPC is not actively looking for a connection. A transition from '1' to '0' indicates a potential connection has been found.
1	TCPC is looking for a connection (toggling as a DRP or looking for a connection as sink/source only condition).

Bit 4 – CONRES Connect Result

Value	Description
0	TCPC is presenting Rp.
1	TCPC is presenting Rd.

Bits 3:2 – CC2[1:0] CC2 State
If (TCPC_RCTL.CC2=Rp) or (ConnectResult=0):

Value	Name	Description
0	CC2_SRC_OPEN	SRC.Open (Open, Rp)
1	CC2_SRC_RA	SRC.Ra (below maximum VRa)
2	CC2_SRC_RD	SRC.Rd (within the VRd range)
3	CC2_SRC_RESERVED	Reserved

If (TCPC_RCTL.CC2 = Rd) or (ConnectResult=1):

Value	Name	Description
0	CC2_SNK_OPEN	SNK.Open (Open, below maximum VRa)
1	CC2_SNK_DEFAULT	SRC.Default (above minimum VRd-Connect)
2	CC2_SNK_1P5A	SRC.Power1.5 (above minimum VRd-Connect) Detects Rp-1.5A
3	CC2_SNK_3A	SNK.Power3.0 (above minimum VRd-Connect) Detects Rp-3.0A

If TCPC_RCTL.CC2=Ra, this field is set to 0.
If TCPC_RCTL.CC2=Open, this field is set to 0.
This field always returns 0 if (LK4CNX=1).
Otherwise, the returned value depends on TCPC_RCTL.CC2.

Bits 1:0 – CC1[1:0] CC1 State
If (TCPC_RCTL.CC1=Rp) or (ConnectResult=0):

Value	Name	Description
0	CC1_SRC_OPEN	SRC.Open (Open, Rp)
1	CC1_SRC_RA	SRC.Ra (below maximum VRa)

...........continued		
Value	**Name**	**Description**
2	CC1_SRC_RD	SRC.Rd (within the VRd range)
3	CC1_SRC_RESERVED	Reserved

If (TCPC_RCTL.CC1 = Rd) or (ConnectResult=1):

Value	Name	Description
0	CC1_SNK_OPEN	SNK.Open (Open, below maximum VRa)
1	CC1_SNK_DEFAULT	SRC.Default (above minimum VRd-Connect)
2	CC1_SNK_1P5A	SRC.Power1.5 (above minimum VRd-Connect) Detects Rp-1.5A
3	CC1_SNK_3A	SNK.Power3.0 (above minimum VRd-Connect) Detects Rp-3.0A

If TCPC_RCTL.CC1=Ra, this field is set to 0.
If TCPC_RCTL.CC1=Open, this field is set to 0.
This field always returns 0 if (LK4CNX=1).
Otherwise, the returned value depends on TCPC_RCTL.CC1.

70.7.15 TCPC Power Status Register

Name: TCPC_PS
Offset: 0x1E
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
	DBG	INIT	SRCHIV	SRCVBUS	VBUSDETE	VBUS	VCONN	SNKVBUS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – DBG Debug Accessory Connected
Reflects the state of the DebugAccessoryConnected# output if supported

Value	Description
0	No debug accessory connected (default).

Bit 6 – INIT TCPC Initialization

Value	Description
0	TCPC has completed initialization and all registers are valid.
1	TCPC is still performing internal initialization and the only registers ensuring return of correct values are 00h..0Fh.

Bit 5 – SRCHIV Sourcing High Voltage

Value	Description
0	vSafe5V.

Bit 4 – SRCVBUS Sourcing VBUS
This does not control the path. It provides a monitor of the status.

Value	Description
0	Sourcing V_{BUS} is disabled.
1	Sourcing V_{BUS} is enabled.

Bit 3 – VBUSDETE VBUS Present Detection Enabled
Indicates if the TCPC is monitoring for V_{BUS} present or if the circuit has been powered off.

Value	Description
0	V_{BUS} present detection disabled.
1	V_{BUS} present detection enabled (default).

Bit 2 – VBUS VBUS Present

Value	Description
0	V_{BUS} disconnected.
1	V_{BUS} connected.

Bit 1 – VCONN VCONN Present

Value	Description
0	VCONN is not present.

Bit 0 – SNKVBUS Sinking VBUS

Value	Description
0	Sink is disconnected. (Default and if not supported)
1	TCPC is sinking V_{BUS} to the system load.

70.7.16 TCPC Fault Status Register

Name: TCPC_FS
Offset: 0x1F
Reset: 0x80
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
	ALLREGRES	FRCOFVBUS			VBUSOCPF	VBUSOVPF		
Access	R/W	R/W			R/W	R/W		
Reset	1	0			0	0		

Bit 7 – ALLREGRES All Registers Reset To Default

Value	Description
0	The registers have not been reset since the last clear of this bit.
1	The TCPC has reset all registers to their default value. This happens at initial powerup or if an unexpected power reset occurs.

Bit 6 – FRCOFVBUS Force Off VBUS

The TCPC has disconnected V_{BUS} due to TCPC_SIC.FOFVBUS.

Value	Description
0	No fault detected, no action (default and not supported).
1	V_{BUS} source/sink has been forced off due to external fault.

Bit 3 – VBUSOCPF Internal or External OCP VBUS Over-Current Protection Fault

Value	Description
0	Not in an over-current protection state.
1	Over-current fault latched.

Bit 2 – VBUSOVPF Internal or External OVP VBUS Over-Voltage Protection Fault

Value	Description
0	Not in an over-voltage protection state.
1	Over-voltage fault latched.

70.7.17 TCPC Command Register

Name: TCPC_CMD
Offset: 0x23
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
	COMMAND[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – COMMAND[7:0] Command
All values not given here are reserved; must be ignored by the receiver.

Value	Name	Description
0010 0010b	DISABLEVBUSDETECT	Disables Vbus present detection.
0011 0011b	ENABLEVBUSDETECT	Enables Vbus present detection.
0100 0100b	DISABLESINKVBUS	Disables sinking power over V_{BUS}. This COMMAND does not disable TCPC_PS.VBUS detection. The TCPC must clear TCPC_FS.VBUSOCPF and TCPC_FS.VBUSOVPF.
0101 0101b	SINKVBUS	Enables sinking power over V_{BUS} and enable V_{BUS} present detection.
0110 0110b	DISABLESOURCEVBUS	Disables sourcing power over V_{BUS}. The TCPC shall stop reporting TCPC_FS. Internal or External OCP or OVP Faults. This COMMAND does not disable TCPC_PS.VBUS detection.
0111 0111b	SOURCEVBUSDEFAULTVOLTAGE	Enables sourcing vSafe5V over V_{BUS} and enable V_{BUS} present detection. Source must transition to vSafe5V if at a high voltage.
1001 1001b	LOOK4CONNECTION	Starts DRP toggling if TCPC_RCTL.DRP=1b. If TCPC_RCTL.CC1/CC2= 01b start with Rp, if TCPC_RCTL.CC1/CC2 =10b start with Rd. If TCPC_RCTL.CC1/CC2 are not both 01b or 10b, then do not start toggling. The TCPM must issue TCPC_CMD.Look4Connection to enable the TCPC to restart connection detection in cases where TCPC_RCTL contents will not change. An example of this is when a potential connection as a source occurred but was further debounced by the TCPM to find the sink disconnected. In this case, a source only or DRP should go back to its Unattached.Src state. This would result in TCPC_RCTL staying the same.

70.7.18 TCPC Device Capabilities 1 Register

Name: TCPC_DCP1
Offset: 0x24
Reset: 0x0000
Property: Read-only, Write-once

Bit	15	14	13	12	11	10	9	8
		VBUSOCPR	VBUSOVPR	BLDDCH	FRCDCH	VBUSMSRAL	SRCRES[1:0]	
Access								
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ROLES[2:0]			SOPDBG	SRCVCN	SNKVBUS	SRCHVBUS	SRCVBUS
Access								
Reset	0	0	0	0	0	0	0	0

Bit 14 – VBUSOCPR VBUS OCP Reporting
Support for both TCPC_FS.VBUSOCPF and TCPC_FCTL.VBUSOCPF implemented.

Value	Description
0	V_{BUS} OCP is not reported by the TCPC.
1	V_{BUS} OCP is reported by the TCPC.

Bit 13 – VBUSOVPR VBUS OVP Reporting
Support for both TCPC_FS.VBUSOVPF and TCPC_FCTL.VBUSOVPF implemented.

Value	Description
0	V_{BUS} OVP is not reported by the TCPC.
1	V_{BUS} OVP is reported by the TCPC.

Bit 12 – BLDDCH Bleed Discharge

Value	Description
0	Enable Watchdog Timer not implemented.
1	Bleed discharge is implemented in the TCPC.

Bit 11 – FRCDCH Force Discharge

Value	Description
0	No force discharge implemented in TCPC.
1	force discharge is implemented in the TCPC.

Bit 10 – VBUSMSRAL VBUS Measurement and Alarm Capable
Support for VBUS_VOLTAGE, VBUS_VOLTAGE_ALARM_HI_CFG, VBUS_VOLTAGE_ALARM_LO_CFG implemented.

Value	Description
0	No V_{BUS} voltage measurement nor V_{BUS} alarms.
1	V_{BUS} voltage measurement and V_{BUS} alarms.

Bits 9:8 – SRCRES[1:0] Source Resistor Supported
Rp values which may be configured by the TCPM via TCPC_RCTL:

Value	Name	Description
0	RES_RPDEF	Rp default only
1	RES_RP1P5	Rp 1.5A and default
2	RES_RP3	Rp 3.0A, 1.5A, and default
3	RES_RESERVED	Reserved

Bits 7:5 – ROLES[2:0] Roles Supported

Value	Name	Description
0	ROLE_SRCSNK	USB Type-C Port Manager can configure the Port as Source only or Sink only (not DRP)
1	ROLE_SRC	Source only
2	ROLE_SNK	Sink only
3	ROLE_SNK_ACC	Sink with accessory support
4	ROLE_DRP	DRP only
5	ROLE_ALL	Source, Sink, DRP, Adapter/Cable all supported
6	ROLE_SRCSNKDRP	Source, Sink, DRP
7	ROLE_INVALID	Not valid

Bit 4 – SOPDBG SOP'_DBG/SOP''_DBG Support
This function is not supported.

Bit 3 – SRCVCN Source VCONN

Value	Description
0	TCPC is not capable of switching VCONN.
1	TCPC is capable of switching VCONN.

Bit 2 – SNKVBUS Sink VBUS
Support for TCPC_PS.SNKVBUS, TCPC_CMD.SinkVbus, and TCPC_CMD.DisableSinkVbus implemented.

Value	Description
0	TCPC is not capable of controlling the sink path to the system load.
1	TCPC is capable of controlling the sink path to the system load.

Bit 1 – SRCHVBUS Source High Voltage VBUS
Support for VBUS_VOLTAGE, TCPC_PS.SRCHIV, and TCPC_CMD.SourceVbusHighVoltage implemented.. DEVICE_CAPABILITIES_1.VBUS Measurement and Alarm Capable must be set to 1b if Source High Voltage VBUS is enabled.

Value	Description
0	TCPC is not capable of controlling the source high voltage path to V_{BUS}.
1	TCPC is capable of controlling the source high voltage path to V_{BUS}. Note:

Bit 0 – SRCVBUS Source VBUS
Support for TCPC_PS.SRCVBUS, TCPC_CMD.SourceVbusDefaultVoltage, TCPC_CMD.DisableSourceVbus, TCPC_CMD.EnableVbusDetect and TCPC_CMD.DisableVbusDetect implemented.

Value	Description
0	TCPC is not capable of controlling the source path to V_{BUS}.
1	TCPC is capable of controlling the source path to V_{BUS}.

70.7.19 TCPC Device Capabilities 2 Register

Name: TCPC_DCP2
Offset: 0x26
Reset: 0x0000
Property: Read-only, Write-once

Bit	15	14	13	12	11	10	9	8
								WDTMR
Access								
Reset								0

Bit	7	6	5	4	3	2	1	0
	SKDSCDET	STPDSCHTH	VBUSVALSB[1:0]		VCPSP[2:0]			VCOFC
Access								
Reset	0	0	0	0	0	0	0	0

Bit 8 – WDTMR Watchdog Timer

Value	Description
0	Enable Watchdog Timer not implemented.
1	Enable Watchdog Timer implemented.

Bit 7 – SKDSCDET Sink Disconnect Detection

Value	Description
0	VBUS_SINK_DISCONNECT_THRESHOLD not implemented (default: Use TCPC_PS.VBUS=0b to indicate a Sink disconnect).
1	VBUS_SINK_DISCONNECT_THRESHOLD implemented.

Bit 6 – STPDSCHTH Stop Discharge Threshold
This function is not supported.

Bits 5:4 – VBUSVALSB[1:0] VBUS Voltage Alarm LSB
This function is not supported.

Bits 3:1 – VCPSP[2:0] VCONN Power Supported
This function is not supported.

Bit 0 – VCOFC VCONN Overcurrent Fault Capable
This function is not supported.

70.7.20 TCPC Standard Input Capabilities Register

Name: TCPC_SIC
Offset: 0x28
Reset: 0x00
Property: Read-only, Write-once

Bit	7	6	5	4	3	2	1	0
						VBUSEOVF	VBUSEOCF	FOFVBUS
Access								
Reset						0	0	0

Bit 2 – VBUSEOVF VBUS External Over-Voltage Fault

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 1 – VBUSEOCF VBUS External Over-Current Fault

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 0 – FOFVBUS Force Off VBUS (Source or Sink)

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

70.7.21 TCPC Standard Output Capabilities Register

Name: TCPC_SOC
Offset: 0x29
Reset: 0x00
Property: Read/Write

Read-only, Write-once

Bit	7	6	5	4	3	2	1	0
		DBGAI	VBUSPM	AAAI	ACI	MXCFGCTL	CNXPR	CNROR
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – DBGAI Debug Accessory Indicator

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 5 – VBUSPM VBUS Present Monitor

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 4 – AAAI Audio Adapter Accessory Indicator

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 3 – ACI Active Cable Indicator

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 2 – MXCFGCTL MUX Configuration Control

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

Bit 1 – CNXPR Connection Present
Controlled by the TCPM.

Value	Description
0	No connection.
1	Connection.

Bit 0 – CNROR Connector Orientation

Value	Description
0	Not present in TCPC.
1	Present in TCPC.

70.7.22 TCPC Control Register

Name: TCPC_CR
Offset: 0x80
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	WAKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bits 31:8 – WAKEY[23:0] Register Write Access Key

Value	Name	Description
`0x544343`	PASSWD	Writing any other value in this field aborts the write operation.

Bit 0 – SWRST Software Reset

Value	Description
`0`	No effect.
`1`	Resets the TCPC. A software-triggered hardware reset of the TCPC interface is performed.

70.7.23 TCPC Set Status Register

Name: TCPC_SSR
Offset: 0x84
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	FAULT_STATUS[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	POWER_STATUS[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CC_STATUS[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:24 – FAULT_STATUS[7:0] Fault Status Register Set

Value	Description
0	No effect.
1	Sets the corresponding bit in TCPC_FS and the FLT_ST bit in TCPC_AL.

Bits 23:16 – POWER_STATUS[7:0] Power Status Register Set

Value	Description
0	No effect.
1	Sets the corresponding bit in TCPC_PS and the PWR_ST bit in TCPC_AL.

Bits 15:8 – CC_STATUS[7:0] CC Status Register Set

Value	Description
0	No effect.
1	Sets the corresponding bit in TCPC_CCS and the CC_ST bit in TCPC_AL.

70.7.24 TCPC Clear Status Register

Name: TCPC_CSR
Offset: 0x88
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	POWER_STATUS[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CC_STATUS[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:16 – POWER_STATUS[7:0] Power Status Register Clear

Value	Description
0	No effect.
1	Clears the corresponding bit in TCPC_PS.

Bits 15:8 – CC_STATUS[7:0] CC Status Register Clear

Value	Description
0	No effect.
1	Clears the corresponding bit in TCPC_CCS.

70.7.25 TCPC Set Alert Register

Name: TCPC_SAR
Offset: 0x8C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ALERT[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ALERT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 15:0 – ALERT[15:0] Alert Register Set

Value	Description
0	No effect.
1	Sets the corresponding bit in TCPC_AL.

70.7.26 TCPC USB Phy Control Register

Name: TCPC_UPC
Offset: 0xA0
Reset: 0x0000
Property: Read/Write

8-bit or 16-bit access to part of this register is permitted.

Bit	31	30	29	28	27	26	25	24
			BCDETE	BCVSRCE	BCDETSEL	BCIDPSRCE	DMPDFE	DMPDFD
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RDIPSEL[1:0]			TCIDCTSEL[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 29 – BCDETE Battery Charging Detection Enable

Value	Description
0	Battery charging detection is disabled.
1	Battery charging detection is enabled. • If BCDETSEL is set to 0, DM VDAT comparator and IDM_SINK current source are enabled, and CHG_DET output can be read in TCPC_UPS.CHGDCP. • If BCDETSEL is set to 1, DP VDAT comparator and IDP_SINK current source are enabled, and DCP_DET output can be read in TCPC_UPS.CHGDCP.

Bit 28 – BCVSRCE Battery Charging Voltage Source Enable

Value	Description
0	Battery charging VDP_SRC and VDM_SRC voltage sources are disabled.
1	Battery charging voltage source enable: • If BCDETSEL is set to 0, VDP_SRC voltage source is enabled. • If BCDETSEL is set to 1, VDM_SRC voltage source is enabled.

Bit 27 – BCDETSEL Battery Charging Detection Selection

Value	Description
0	Battery charging primary detection is selected.
1	Battery charging secondary detection is selected.

Bit 26 – BCIDPSRCE Battery Charging IDP Source Enable

Value	Description
0	Battery charging IDP current source is disabled.
1	Battery charging IDP current source is enabled for data contact detection.

Bit 25 – DMPDFE DM Pull Down Force Enable

Value	Description
0	DM pull down is not forced to the active state by the TCPC.
1	DM pull down is forced to the active state by the TCPC for battery charging data contact detection.

Bit 24 – DMPDFD DM Pull Down Force Disable

Value	Description
0	DM pull down is not forced to the inactive state by the TCPC.
1	DM pull down is forced to the inactive state by the TCPC.

Bits 13:12 – RDIPSEL[1:0] Type-C Rd or Ip Pull up Current Selection

Value	Name	Description
0	IP_OFF	Type-C pull up current source Ip is off, Rd pull down is selected
1	IP_0P5	Type-C pull up current source Ip is advertising 0.5A as V_{BUS} sourcing capability
2	IP_1P5	Reserved
3	IP_3P0	Reserved

Bits 10:8 – TCIDCTSEL[2:0] Type-C or ID Comparator Threshold Selection

Value	Name	Description
0	THRESHOLD_0	Type-C cell power down
1	THRESHOLD_1	Reserved
2	THRESHOLD_2	Type-C VRa Threshold
3	THRESHOLD_3	Reserved
4	THRESHOLD_4	Type-C VRd-USB Threshold
5	THRESHOLD_5	Reserved
6	THRESHOLD_6	Type-C VRd-1.5 Threshold, Type-C VRd Threshold
7	THRESHOLD_7	Reserved

70.7.27 TCPC USB Phy Status Register

Name: TCPC_UPS
Offset: 0xA4
Reset: 0x0000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CC2RDT	CC1ID	CHGDCP	DM	DP
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – CC2RDT Type-C CC2 Comparator or Rd Trim comparator
Comparator output for Type-C CC2 voltage detection or for Rd trimming as selected via TCPC_UPC.RDSEL.

Value	Description
0	CC2 output is low. Voltage is below the selected TCUPC.TCIDCTSEL threshold for Power Source Detection at Sink and is above the selected TCPC_UPC.TCIDCTSEL threshold for Sink Detection at Power Source.
1	CC2 output is high. Voltage is above the selected TCUPC.TCIDCTSEL threshold for Power Source Detection at Sink and is below the selected TCPC_UPC.TCIDCTSEL threshold for Sink Detection at Power Source.

Bit 3 – CC1ID Type-C CC1 Comparator or ID comparator
Comparator output for Type-C CC1 voltage detection.

Value	Description
0	CC1 output is low. Voltage is below the selected TCPC_UPC.TCIDCTSEL threshold for Power Source Detection at Sink and is above the selected TCPC_UPC.TCIDCTSEL threshold for Sink Detection at Power Source.
1	CC1 output is high. Voltage is above the selected TCPC_UPC.TCIDCTSEL threshold for Power Source Detection at Sink and is below the selected TCPC_UPC.TCIDCTSEL threshold for Sink Detection at Power Source.

Bit 2 – CHGDCP Charging Port Detection, Dedicated Charging Port Detection
Comparator output of the primary or secondary detection as selected via TCPC_UPC.BCDETSEL, valid if TCPC_UPC.BCDETE has been set.

Value	Description
0	No charging port detected • If TCPC_UPC.BCDETSEL is set to 0, the primary detection CHG_DET output is low. • If TCPC_UPC.BCDETSEL is set to 1, the secondary detection DCP_DET output is low.

Value	Description
1	Charging port detected • If TCPC_UPC.BCDETSEL is set to 0, the primary detection CHG_DET output is high. • If TCPC_UPC.BCDETSEL is set to 1, the secondary detection DCP_DET output is high.

Bit 1 – DM USB Differential line Minus
USB DM line state, used in the battery charger type determination to distinguish ACA with FS B-device from ACA with LS B-device.

Value	Description
0	DM is low.
1	DM is high.

Bit 0 – DP USB Differential line Plus
USB DP line state, used for the battery charging data contact detect and ACA-Dock type determination.

Value	Description
0	DP is low.
1	DP is high.

71. USB 2.0 PHY

71.1 Description

The USB 2.0 PHY integrates high-speed, full-speed and low-speed (Host mode only) termination and signal switching. With a single resistor, it requires minimal external components.

The USB 2.0 PHY is compliant with the Universal Serial Bus Specification, Revision 2.0, the USB Battery Charging Specification, Revision 1.2, the Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0, and the UTMI+ Specification, Revision 1.0 (Level 3).

71.2 Embedded Characteristics

- USB 2.0 Features
  - Fully integrates 45-ohm termination, 1.5-Kohm pull-up and 15-Kohm pull-down resistors, with support for independent control of the pull-down resistors
  - Supports 480 Mbps high-speed, 12 Mbps full-speed, and 1.5 Mbps low-speed (Host mode only) data transmission rates
  - Supports suspend, sleep, resume, and remote wakeup operations
  - Supports USB2.0 test modes
- USB Type-C Features
  - Mode configuration, supports Upstream Facing Port (UFP) and Downstream Facing Port (DFP)
  - Supports cable orientation detection
  - Supports default USB Power DFP Advertisement
  - Supports Type-C current mode detection (UFP)
  - Supports power collapse
- Compliance with Standards
  - Universal Serial Bus Specification, Revision 2.0, with a High-Speed (HS) Transmit to Receive Inter-Packet Delay requirement of 38 bit-times provided by the UTMI specification version 1.05.
  - Battery Charging Specification, Revision 1.2, USB Implementers Forum, Inc. ACA (Accessory Charger Adaptor) functions are not supported.
  - Embedded Host Supplement to the USB Revision 2.0 Specification, Revision 2.0, USB
  - UTMI+ Specification, Revision 1.0 (Level 3)

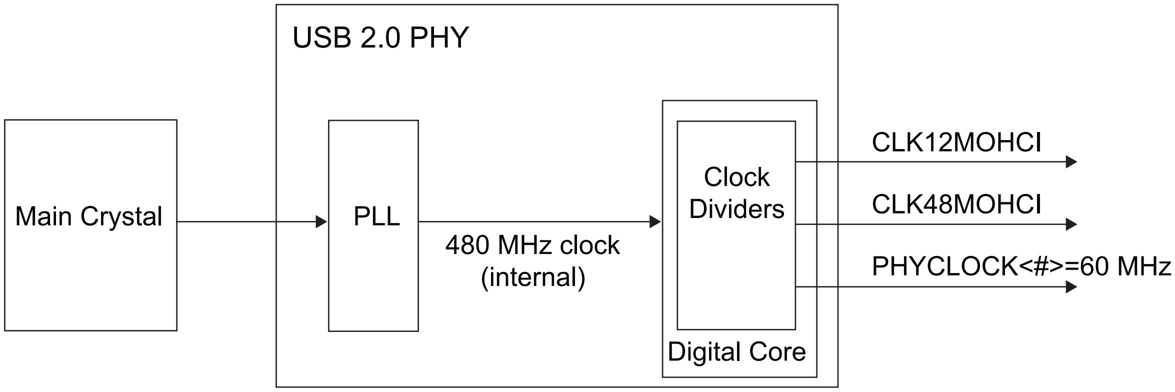
71.3 PHY Setup Procedure

The reference clock accepts 16, 20, 24 and 32 MHz main crystal input frequencies.

To start USB operations, apply the following sequence:

1. Enable the main crystal in PMC_MOR.
2. Set the main crystal frequency in PMC_XTALF to define the multiplier.
3. Release PHY reset in RSTC_GRSTR for each PHY (A, B, C) to be used.
4. Wait for 45 µs before any USB operation.

Figure 71-1. USB 2.0 PHY Clock Generation for Each USB Transceiver



71.4 Typical Connection

Figure 71-2. USB 2.0 PHY and IOs

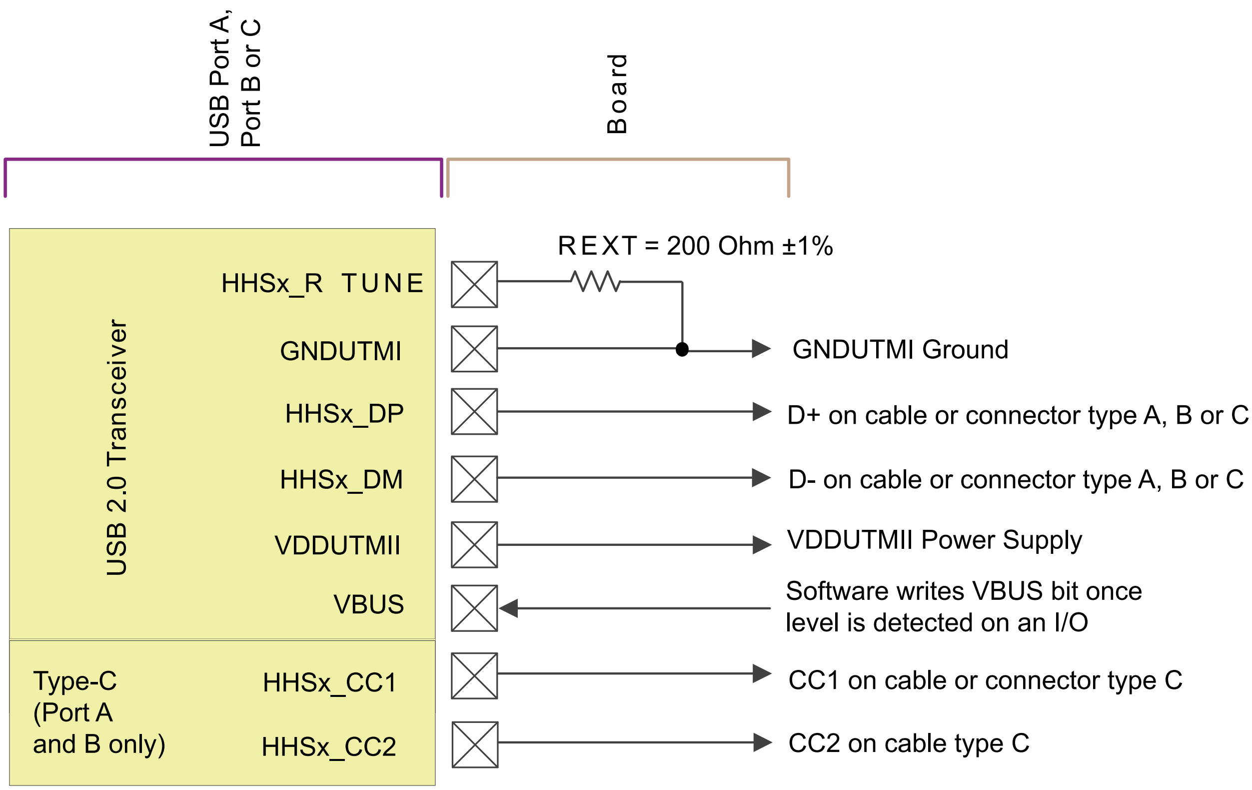


Figure 71-3. USB Type-C Cable Usage for USB2.0

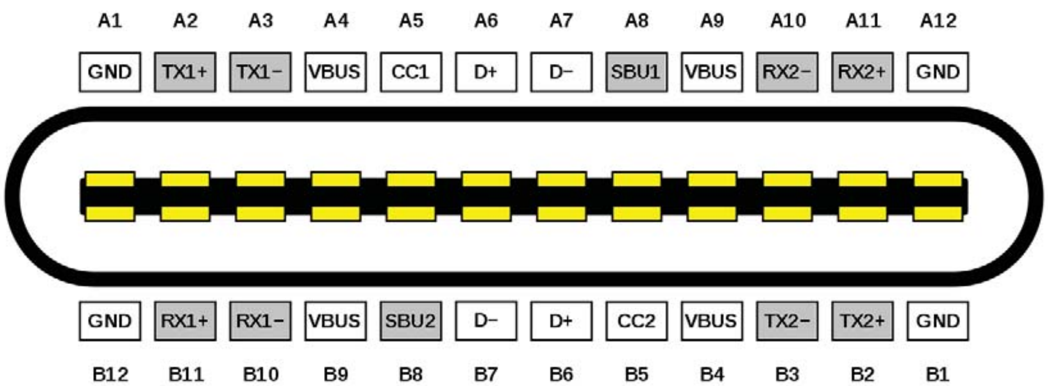
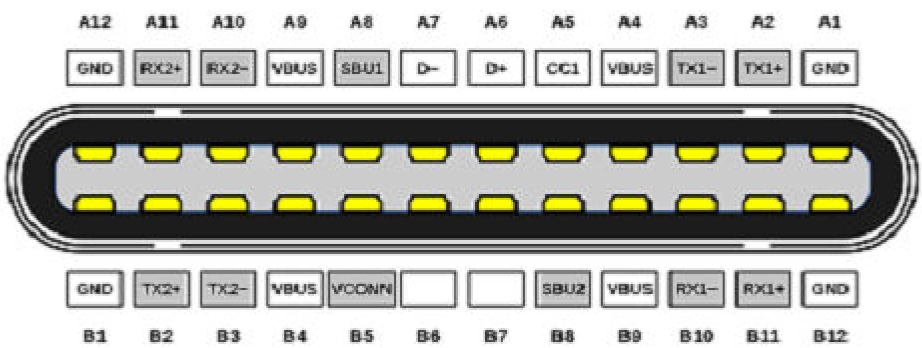


Figure 71-4. USB Type-C Connector Usage for USB2.0



71.5 Product Dependencies

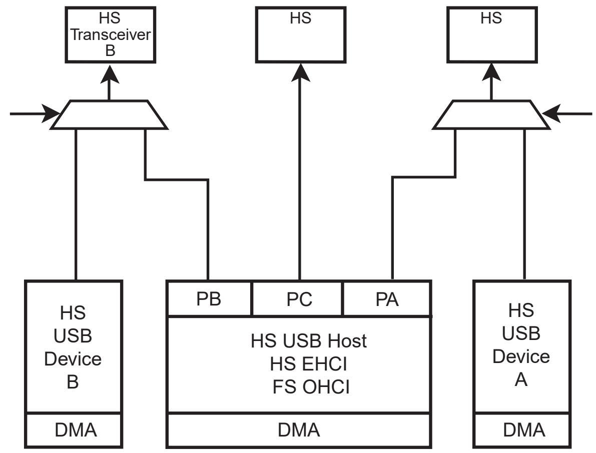
71.5.1 I/O Lines

HHSx_DPs and HHSx_DMs(x=A, B, C) are not controlled by PIO controllers. The embedded USB High Speed physical transceivers are controlled by the USB host controller.

71.5.2 UTMI Transceiver Sharing

The high-speed USB Host ports A and B are shared with the high-speed USB Device ports A and B, respectively . The selection between Host ports A and B and USB Device A and B to drive, respectively, transceivers A and B is controlled by the UDPHS Enable bit (EN_UDPHS) located in the register UDPHS_CTRL. USB port C is not shared and is the only port that drives transceiver C.

Figure 71-5. USB Selection



71.5.3 VBUS Management

Any I/O configured as an input with level detection can be used to monitor the VBUS signal. Software has to generate an interrupt on level detection and write the relevant VBUS bit in SFR_UTMI0R0, SFR_UTMI0R1 or SFR_UTMI0R2, respectively for USB ports A, B or C, in order to indicate to the USB controller the VBUS signal is valid.

72. USB Device High Speed Port (UDPHS)

72.1 Description

The USB Device High Speed Port (UDPHS) is compliant with the Universal Serial Bus (USB), rev 2.0 High Speed device specification.

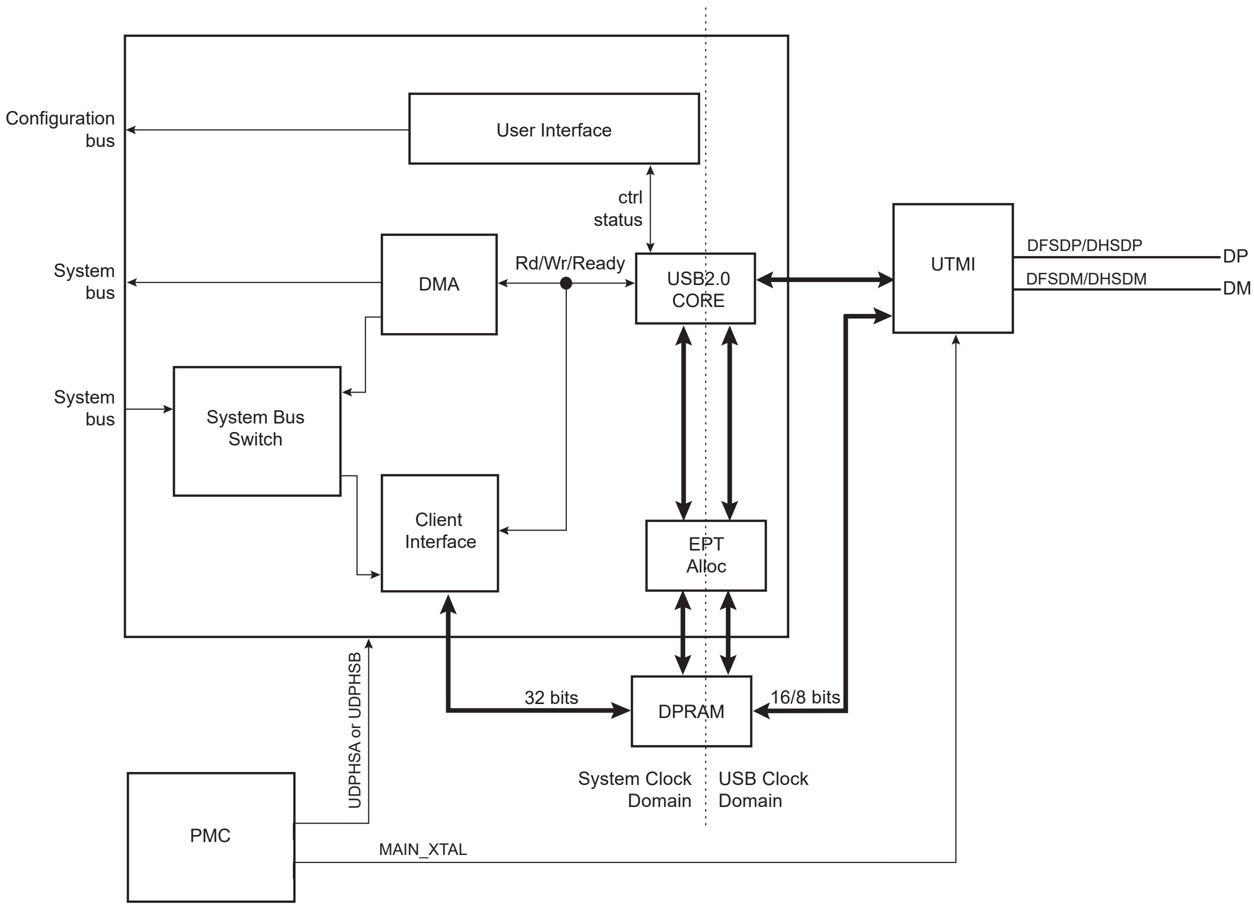
Each endpoint can be configured in one of several USB transfer types. It can be associated with one, two or three banks of a Dual-port RAM used to store the current data payload. If two or three banks are used, one DPR bank is read or written by the processor, while the other is read or written by the USB device peripheral. This feature is mandatory for isochronous endpoints.

72.2 Embedded Characteristics

- 1 High-Speed Device
- 1 UTMI Transceiver Shared Between Host and Device for Each UDPHS Instance
- USB v2.0 High Speed (480 Mbits/s) Compliant
- 16 Endpoints up to 1024 Bytes
- Embedded Dual-port RAM for Endpoints
- Suspend/Resume Logic (Command of UTMI)
- Up to Three Memory Banks for Endpoints (Not for Control Endpoint)
- 16448 bytes of DPRAM

72.3 Block Diagram

Figure 72-1. UDPHS Block Diagram



72.4 Typical Connection

Refer to the figure “USB 2.0 PHY and IOs” in the section “USB 2.0 PHY”.

72.5 Product Dependencies

72.5.1 Power Management

The UDPHS is not continuously clocked.

To use the UDPHS, the programmer must first enable the UDPHS clock in the Power Management Controller (PMC).

However, if the application does not require UDPHS operations, the UDPHS clock can be stopped when not needed and restarted later.

72.5.2 Interrupt Sources

The UDPHS interrupt line is connected on one of the internal sources of the interrupt controller. Using the UDPHS interrupt requires the interrupt controller to be programmed first.

72.6 Functional Description

72.6.1 USB V2.0 High Speed Device Port Introduction

The USB V2.0 High Speed Device Port provides communication services between host and attached USB devices. Each device is offered with a collection of communication flows (pipes) associated with each endpoint. Software on the host communicates with a USB Device through a set of communication flows.

72.6.2 USB V2.0 High Speed Transfer Types

A communication flow is carried over one of four transfer types defined by the USB device.

A device provides several logical communication pipes with the host. To each logical pipe is associated an endpoint. Transfer through a pipe belongs to one of the four transfer types:

- Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- Bulk Data Transfers: Generated or consumed in relatively large burst quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers.)

As indicated below, transfers are sequential events carried out on the USB bus.

Endpoints must be configured according to the transfer type they handle.

Table 72-1. USB Communication Flow

Transfer	Direction	Bandwidth	Endpoint Size	Error Detection	Retrying
Control	Bidirectional	Not ensured	8, 16, 32, 64	Yes	Automatic
Isochronous	Unidirectional	Ensured	8–1024	Yes	No
Interrupt	Unidirectional	Not ensured	8–1024	Yes	Yes
Bulk	Unidirectional	Not ensured	8–512	Yes	Yes

72.6.3 USB Transfer Event Definitions

A transfer is composed of one or several transactions as shown in the table below.

Table 72-2. USB Transfer Events

<table>
<tr><th colspan="2">Transfer</th><th rowspan="2">Transaction</th></tr>
<tr><th>Direction</th><th>Type</th></tr>
<tr><td>CONTROL (bidirectional)</td><td>Control Transfer (1)</td><td>• Setup transaction → Data IN transactions → Status OUT transaction
• Setup transaction → Data OUT transactions → Status IN transaction
• Setup transaction → Status IN transaction</td></tr>
<tr><td rowspan="3">IN (device toward host)</td><td>Bulk IN Transfer</td><td>• Data IN transaction → Data IN transaction</td></tr>
<tr><td>Interrupt IN Transfer</td><td>• Data IN transaction → Data IN transaction</td></tr>
<tr><td>Isochronous IN Transfer (2)</td><td>• Data IN transaction → Data IN transaction</td></tr>
</table>

...........continued

Transfer		Transaction
Direction	**Type**	
OUT (host toward device)	Bulk OUT Transfer	• Data OUT transaction → Data OUT transaction
	Interrupt OUT Transfer	• Data OUT transaction → Data OUT transaction
	Isochronous OUT Transfer [(2)]	• Data OUT transaction → Data OUT transaction

Notes:
1. Control transfer must use endpoints with one bank and can be aborted using a stall handshake.
2. Isochronous transfers must use endpoints configured with two or three banks.

An endpoint handles all transactions related to the type of transfer for which it has been configured.

Table 72-3. UDPHS Endpoint Description

Endpoint #	Mnemonic	Nb Banks	DMA	High Bandwidth	Max. Endpoint Size	Endpoint Type
0	EPT_0	1	N	N	64	Control
1	EPT_1	3	Y	Y	1024	Ctrl/Bulk/Iso[(1)]/Interrupt
2	EPT_2	3	Y	Y	1024	Ctrl/Bulk/Iso[(1)]/Interrupt
3	EPT_3	2	Y	N	1024	Ctrl/Bulk/Iso[(1)]/Interrupt
4	EPT_4	2	Y	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
5	EPT_5	2	Y	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
6	EPT_6	2	Y	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
7	EPT_7	2	Y	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
8	EPT_8	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
9	EPT_9	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
10	EPT_10	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
11	EPT_11	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
12	EPT_12	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
13	EPT_13	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
14	EPT_14	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt
15	EPT_15	1	N	N	512	Ctrl/Bulk/Iso[(1)]/Interrupt

Note:
1. In Isochronous (Iso) mode, it is preferable that the high bandwidth capability is available.

The size of the internal DPRAM is 16448 bytes, covering the memory need for the seven endpoints, hence enabling static allocation of the memory for all endpoints.

Suspend and resume are automatically detected by the UDPHS device, which notifies the processor by raising an interrupt.

72.6.4 USB V2.0 High Speed BUS Transactions

Each transfer results in one or more transactions over the USB bus.

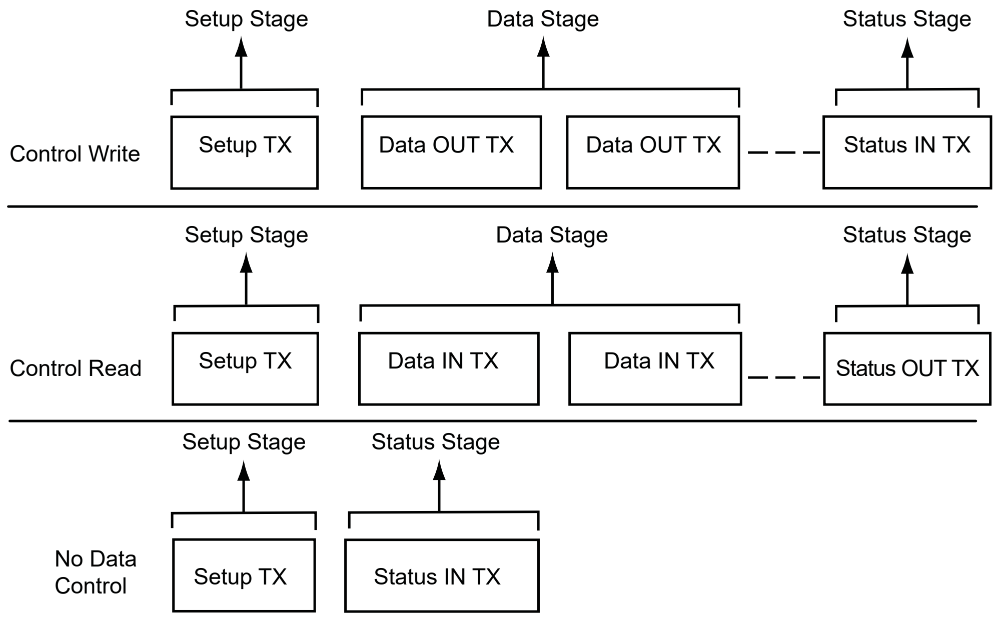
Five types of transaction flow across the bus in packets:

1. Setup Transaction

2. Data IN Transaction
3. Data OUT Transaction
4. Status IN Transaction
5. Status OUT Transaction

A status IN or OUT transaction is identical to a data IN or OUT transaction.

Figure 72-2. Control Read and Write Sequences



72.6.5 Endpoint Configuration

The endpoint 0 is always a control endpoint, it must be programmed and active in order to be enabled when the End Of Reset interrupt occurs.

To configure the endpoints:

- Fill the configuration register (UDPHS_EPTCFG) with the endpoint size, direction (IN or OUT), type (CTRL, Bulk, IT, ISO) and the number of banks.
- Fill the number of transactions (NB_TRANS) for isochronous endpoints.

Note: For control endpoints the direction has no effect.

- Verify that the EPT_MAPD flag is set. This flag is set if the endpoint size and the number of banks are correct for this endpoint.
- Configure the endpoint control flags and enable them in UDPHS_EPTCTLENBx as described in UDPHS_EPTCTLDISx.

Control endpoints can generate interrupts and use only 1 bank.

All endpoints (except endpoint 0) can be configured either as Bulk, Interrupt or Isochronous. See the table USB Transfer Events.

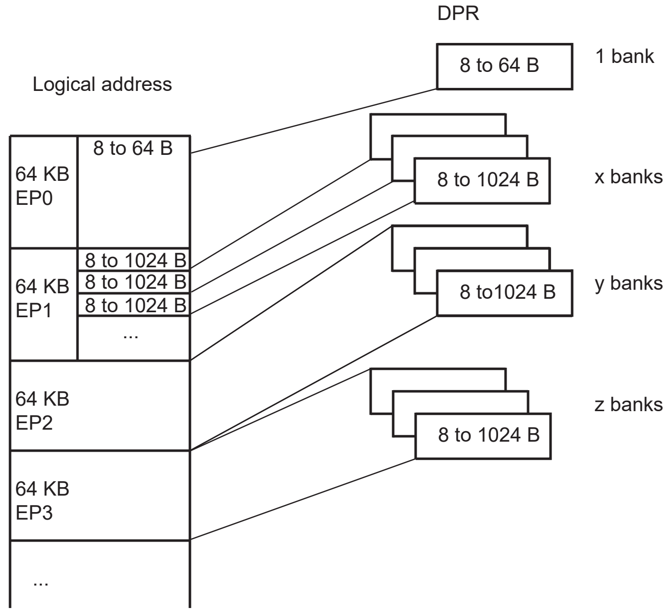
The maximum packet size they can accept corresponds to the maximum endpoint size.

Note: The endpoint size of 1024 is reserved for isochronous endpoints.

The application has access to the physical block of DPR reserved for the endpoint through a 64-Kbyte logical address space.

The physical block of DPR reserved for the endpoint is remapped all along the 64-Kbyte logical address space. The application can write a 64-Kbyte buffer linearly.

Figure 72-3. Logical Address Space for DPR Access



Configuration examples of UDPHS_EPTCTLDISx for Bulk IN endpoint type follow below.

- With DMA
  - AUTO_VALID: Automatically validate the packet and switch to the next bank.
  - EPT_ENABL: Enable endpoint.
- Without DMA:
  - TXRDY: An interrupt is generated after each transmission.
  - EPT_ENABL: Enable endpoint.

Configuration examples of Bulk OUT endpoint type follow below.

- With DMA
  - AUTO_VALID: Automatically validate the packet and switch to the next bank.
  - EPT_ENABL: Enable endpoint.
- Without DMA
  - RXRDY_TXKL: An interrupt is sent after a new packet has been stored in the endpoint FIFO.
  - EPT_ENABL: Enable endpoint.

72.6.6 DPRAM Management

Endpoints can be configured in any order.

Disabling an endpoint, by writing a one to the Endpoint Disable bit in the UDPHS Endpoint Control Disable Register (UDPHS_EPTCTLDISx.EPT_DISABL), does not reset its configuration:

- Endpoint Banks (UDPHS_EPTCFGx.BK_NUMBER)
- Endpoint Size (UDPHS_EPTCFGx.EPT_SIZE)
- Endpoint Direction (UDPHS_EPTCFGx.EPT_DIR)
- Endpoint Type (UDPHS_EPTCFGx.EPT_TYPE)

Notes:
1. There is no way the data of the endpoint 0 can be lost (except if it is de-allocated) as the memory allocation and de-allocation may affect only higher endpoints.
2. Deactivating then reactivating the same endpoint with the same configuration only modifies temporarily the controller DPRAM pointer and size for this endpoint. Nothing changes in the DPRAM, higher endpoints seem not to have been moved and their data is preserved as far as nothing has been written or received into them while changing the allocation state of the first endpoint.
3. When the user writes a value different from zero to the UDPHS_EPTCFGx.BK_NUMBER field, the Endpoint Mapped bit (UDPHS_EPTCFGx.EPT_MAPD) is set only if the configured size and number of banks are correct as compared to the endpoint maximal allowed values and to the maximal FIFO size (i.e., the DPRAM size). The UDPHS_EPTCFGx.EPT_MAPD value does not consider memory allocation conflicts.

72.6.7 Transfer With DMA

USB packets of any length may be transferred when required by the UDPHS device. These transfers always feature sequential addressing.

Packet data system bus bursts may be locked on a DMA buffer basis for drastic overall system bus bandwidth performance boost with paged memories. These clock-cycle consuming memory row (or bank) changes will then likely not occur, or occur only once instead of several times, during a single big USB packet DMA transfer in case another system bus host addresses the memory. The locked bursts result in up to 128-word single-cycle unbroken system bus bursts for bulk endpoints and 256-word single-cycle unbroken bursts for isochronous endpoints.

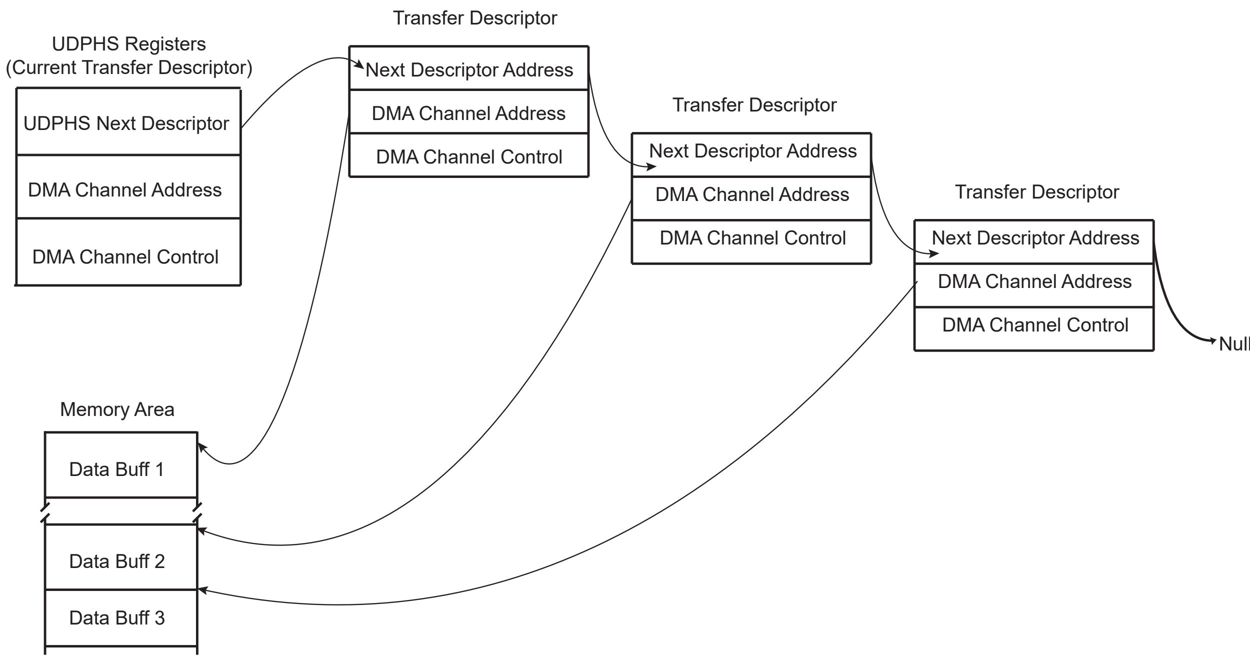
This maximum burst length is then controlled by the lowest programmed USB endpoint size (EPT_SIZE field in the UDPHS_EPTCFGx register) and DMA Size (BUFF_LENGTH field in the UDPHS_DMACONTROLx register).

The USB 2.0 device average throughput may be up to nearly 60 Mbyte/s. Its internal client average access latency decreases as burst length increases due to the 0 wait-state side effect of unchanged endpoints. If at least 0 wait-state word burst capability is also provided by the external DMA system bus clients, each of both DMA system busses need less than 50% bandwidth allocation for full USB 2.0 bandwidth usage at 30 MHz, and less than 25% at 60 MHz.

The UDPHS DMA Channel Transfer Descriptor is described in the section UDPHS DMA Channel Transfer Descriptor.

Note: When debugging, make sure to address the DMA to an SRAM address even if a remap is done.

Figure 72-4. Example of DMA Chained List



72.6.8 Transfer Without DMA

Important: If the DMA is not to be used, it is necessary to disable it, otherwise it can be enabled by previous versions of software without warning. If this should occur, the DMA can process data before an interrupt without knowledge of the user.

The recommended means to disable DMA are as follows:

```
// Reset IP UDPHS
  AT91C_BASE_UDPHS->UDPHS_CTRL &= ~AT91C_UDPHS_EN_UDPHS;
  AT91C_BASE_UDPHS->UDPHS_CTRL |= AT91C_UDPHS_EN_UDPHS;//
With OR without DMA !!!
  for( i=1; i<=((AT91C_BASE_UDPHS->UDPHS_IPFEATURES &
AT91C_UDPHS_DMA_CHANNEL_NBR)>>4); i++ ) {
// RESET endpoint canal DMA:
    // DMA stop channel command
   AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Disable endpoint
  AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLDIS |= 0XFFFFFFFF;
// Reset endpoint config
  AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLCFG = 0;
// Reset DMA channel (Buff count and Control field)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0x02; // NON
STOP command
// Reset DMA channel 0 (STOP)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Clear DMA channel status (read the register for clear it)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS =
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS;
}
```

72.6.9 Handling Transactions with USB V2.0 Device Peripheral

72.6.9.1 Setup Transaction

The setup packet is valid in the DPR while RX_SETUP is set. Once RX_SETUP is cleared by the application, the UDPHS accepts the next packets sent over the device endpoint.

When a valid setup packet is accepted by the UDPHS:

- The UDPHS device automatically acknowledges the setup packet (sends an ACK response)
- Payload data is written in the endpoint
- Sets the RX_SETUP interrupt
- The BYTE_COUNT field in the UDPHS_EPTSTAx register is updated

An endpoint interrupt is generated while RX_SETUP in the UDPHS_EPTSTAx register is not cleared. This interrupt is carried out to the microcontroller if interrupts are enabled for this endpoint.

Thus, firmware must detect RX_SETUP polling UDPHS_EPTSTAx or catching an interrupt, read the setup packet in the FIFO, then clear the RX_SETUP bit in the UDPHS_EPTCLRSTA register to acknowledge the setup stage.

If STALL_SNT was set to 1, then this bit is automatically reset when a setup token is detected by the device. Then, the device still accepts the setup stage (see STALL).

72.6.9.2 NYET

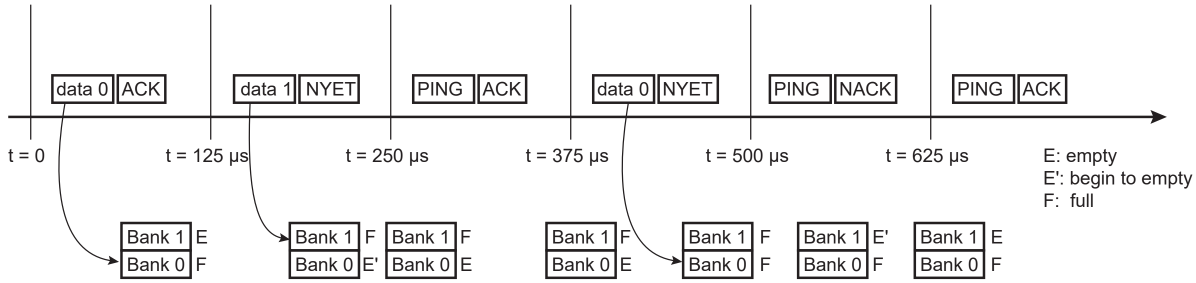
NYET is a High Speed only handshake. It is returned by a High Speed endpoint as part of the PING protocol.

High Speed devices must support an improved NAK mechanism for Bulk OUT and control endpoints (except setup stage). This mechanism allows the device to tell the host whether it has sufficient endpoint space for the next OUT transfer (refer to USB 2.0 spec 8.5.1 NAK Limiting via Ping Flow Control).

The NYET/ACK response to a High Speed Bulk OUT transfer and the PING response are automatically handled by hardware in the UDPHS_EPTCTLx register (except when the user wants to force a NAK response by using the NYET_DIS bit).

If the endpoint responds instead to the OUT/DATA transaction with an NYET handshake, this means that the endpoint accepted the data but does not have room for another data payload. The host controller must return to using a PING token until the endpoint indicates it has space available.

Figure 72-5. NYET Example with Two Endpoint Banks



72.6.9.3 Data IN

- Bulk IN or Interrupt IN
  Data IN packets are sent by the device during the data or the status stage of a control transfer or during an (interrupt/bulk/isochronous) IN transfer. Data buffers are sent packet by packet under the control of the application or under the control of the DMA channel.

  There are three ways for an application to transfer a buffer in several packets over the USB:

  – packet by packet (see Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host) below)
  – 64 Kbytes (see Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host) below)
  – DMA (see Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host) below)
- Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host)
  The application can write one or several banks.

  A simple algorithm can be used by the application to send packets regardless of the number of banks associated to the endpoint.

  Algorithm description for each packet:

  - The application waits for the TXRDY flag to be cleared in the UDPHS_EPTSTAx register before it can perform a write access to the DPR.
  - The application writes one USB packet of data in the DPR through the 64 Kbytes endpoint logical memory window.
  - The application sets TXRDY flag in the UDPHS_EPTSETSTAx register.

  The application is notified that it is possible to write a new packet to the DPR by the TXRDY interrupt. This interrupt can be enabled or masked by setting the TXRDY bit in the UDPHS_EPTCTLENB/ UDPHS_EPTCTLDIS register.

  Algorithm description to fill several packets:

  Using the previous algorithm, the application is interrupted for each packet. It is possible to reduce the application overhead by writing linearly several banks at the same time. The AUTO_VALID bit in the UDPHS_EPTCTLx must be set by writing the AUTO_VALID bit in the UDPHS_EPTCTLENBx register.

  The auto-valid-bank mechanism allows the transfer of data (IN and OUT) without the intervention of the CPU. This means that bank validation (set TXRDY or clear the RXRDY_TXKL bit) is done by hardware.

  - The application checks the BUSY_BANK_STA field in the UDPHS_EPTSTAx register. The application must wait that at least one bank is free.
  - The application writes a number of bytes inferior to the number of free DPR banks for the endpoint. Each time the application writes the last byte of a bank, the TXRDY signal is automatically set by the UDPHS.
  - If the last packet is incomplete (i.e., the last byte of the bank has not been written) the application must set the TXRDY bit in the UDPHS_EPTSETSTAx register.

The application is notified that all banks are free, so that it is possible to write another burst of packets by the BUSY_BANK interrupt. This interrupt can be enabled or masked by setting the BUSY_BANK flag in the UDPHS_EPTCTLENB and UDPHS_EPTCTLDIS registers.

This algorithm must not be used for isochronous transfer. In this case, the ping-pong mechanism does not operate.

A Zero Length Packet can be sent by setting just the TXRDY flag in the UDPHS_EPTSETSTAx register.

- Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host)
  The UDPHS integrates a DMA host controller. This DMA controller can be used to transfer a buffer from the memory to the DPR or from the DPR to the processor memory under the UDPHS control. The DMA can be used for all transfer types except control transfer.

  Example DMA configuration:
  - 1. Program UDPHS_DMAADDRESS x with the address of the buffer that should be transferred.
  - 2. Enable the interrupt of the DMA in UDPHS_IEN.
  - 3. Program UDPHS_ DMACONTROLx:
    – Size of buffer to send: size of the buffer to be sent to the host.
    – END_B_EN: the endpoint can validate the packet (according to the values programmed in the AUTO_VALID and SHRT_PCKT fields of UDPHS_EPTCTLx) (see UDPHS_EPTCTLDISx and the figure Autovalid with DMA).
    – END_BUFFIT: generate an interrupt when the BUFF_COUNT field in the UDPHS DMA Channel Status register (UDPHS_DMASTATUSx) reaches 0.
    – CHANN_ENB: run and stop at end of buffer.

The auto-valid-bank mechanism allows the transfer of data (IN & OUT) without the intervention of the CPU. This means that bank validation (set TXRDY or clear RXRDY_TXKL) is done by hardware.

A transfer descriptor can be used. Instead of programming the register directly, a descriptor should be programmed and the address of this descriptor is then given to UDPHS_DMANXTDSC to be processed after setting the LDNXT_DSC field (Load Next Descriptor Now) in UDPHS_DMACONTROLx register.

The structure that defines this transfer descriptor must be aligned.

Each buffer to be transferred must be described by a DMA Transfer descriptor (see UDPHS DMA Channel Transfer Descriptor). Transfer descriptors are chained. Before executing transfer of the buffer, the UDPHS may fetch a new transfer descriptor from the memory address pointed by the UDPHS_DMANXTDSCx register. Once the transfer is complete, the transfer status is updated in the UDPHS_DMASTATUSx register.

To chain a new transfer descriptor with the current DMA transfer, the DMA channel must be stopped. To do so, INTDIS_DMA and TXRDY may be set in the UDPHS_EPTCTLENBx register. It is also possible for the application to wait for the completion of all transfers. In this case the LDNXT_DSC bit in the last transfer descriptor UDPHS_DMACONTROLx register must be set to 0 and the CHANN_ENB bit set to 1.

Then the application can chain a new transfer descriptor.

The INTDIS_DMA can be used to stop the current DMA transfer if an enabled interrupt is triggered. This can be used to stop DMA transfers in case of errors.

The application can be notified at the end of any buffer transfer (via UDPHS_DMACONTROLx.ENB_BUFFIT).

Figure 72-6. Data IN Transfer for Endpoint with One Bank

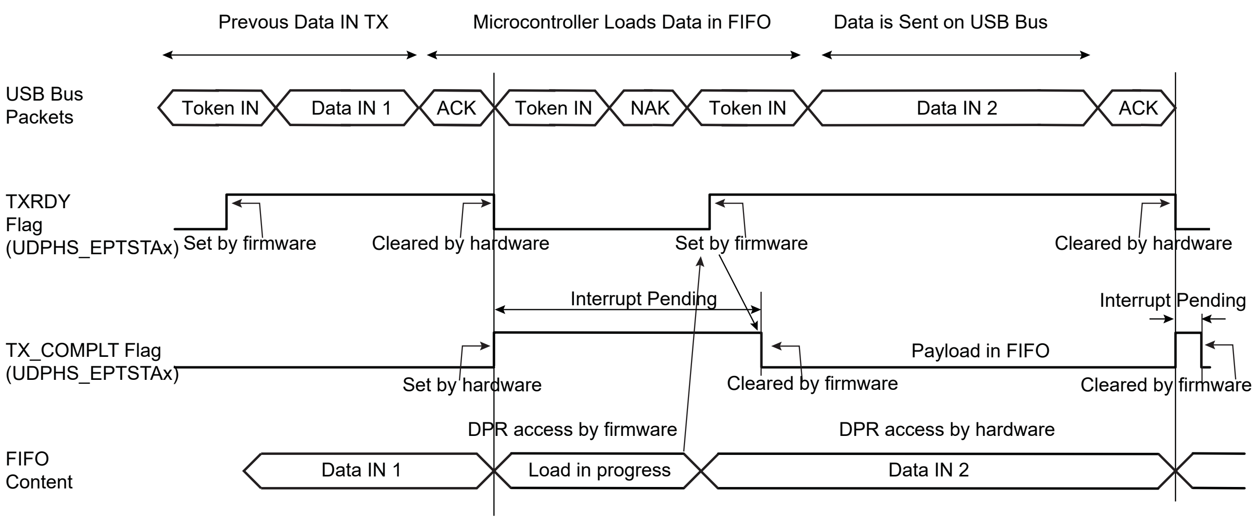


Figure 72-7. Data IN Transfer for Endpoint with Two Banks

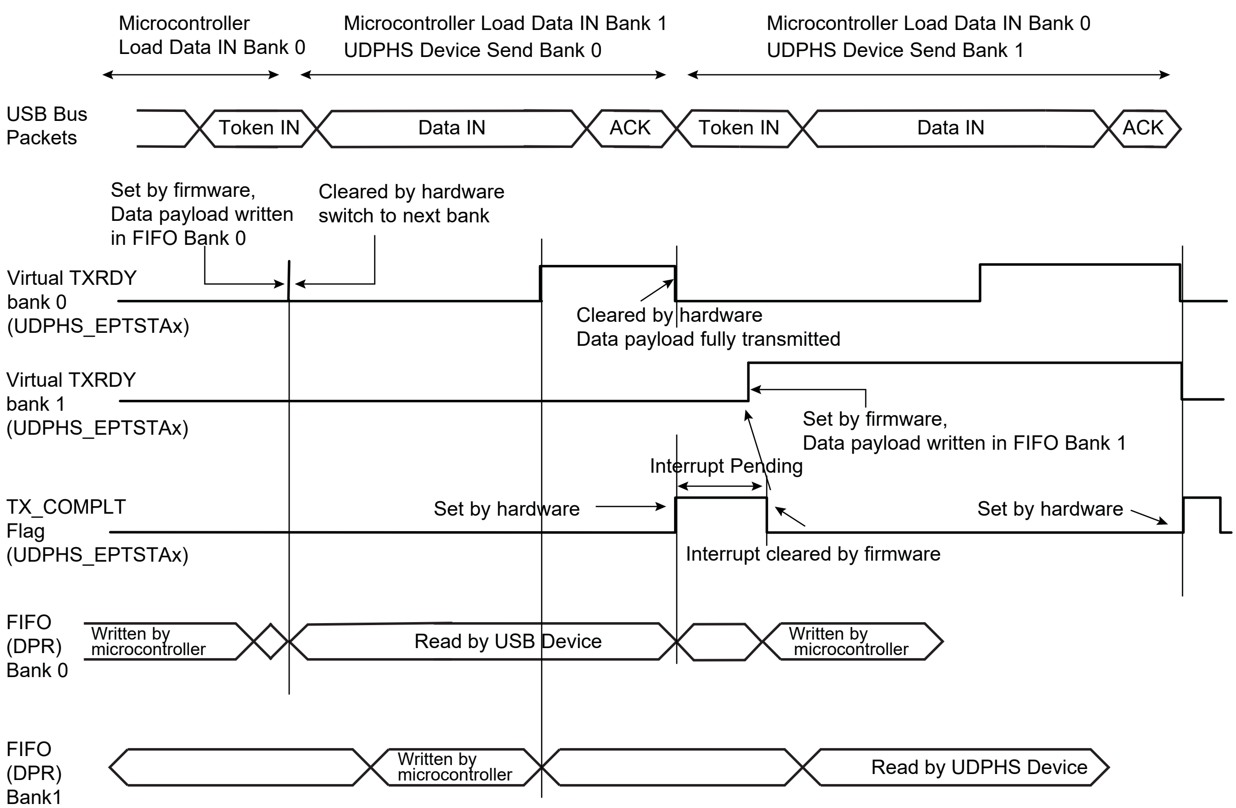
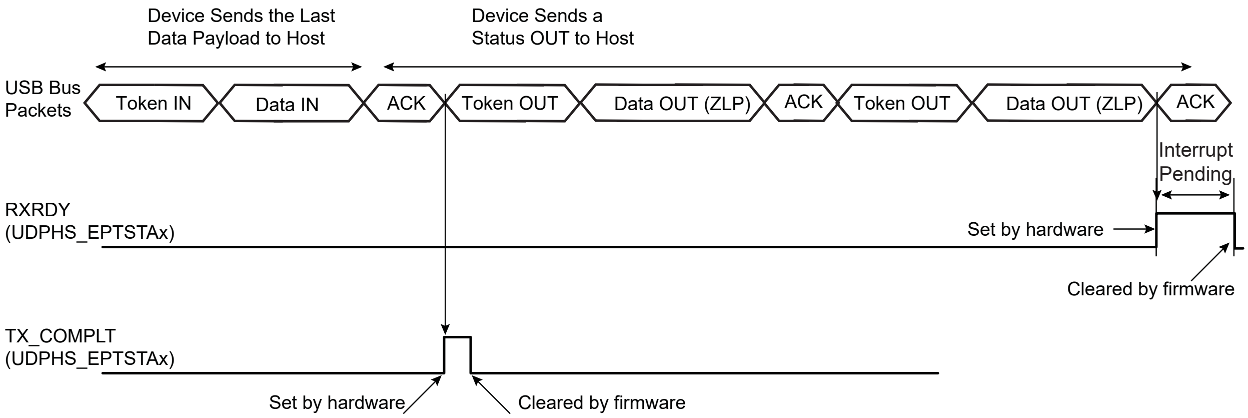
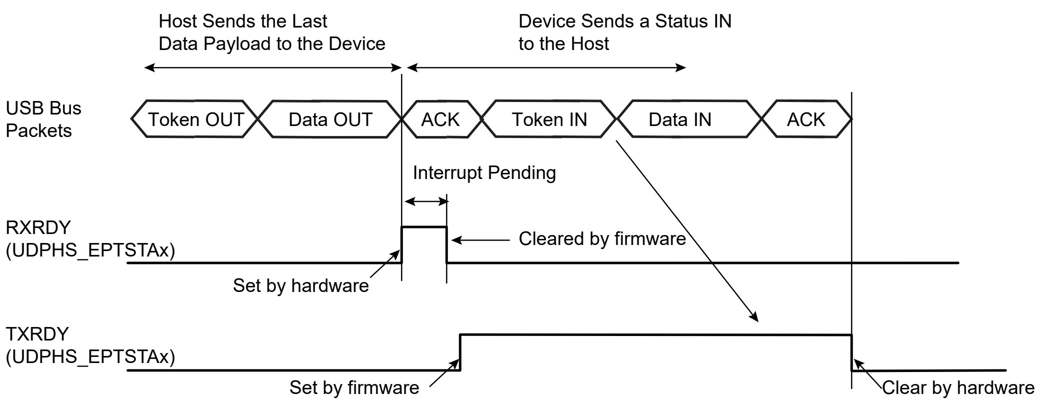


Figure 72-8. Data IN Followed By Status OUT Transfer at the End of a Control Transfer



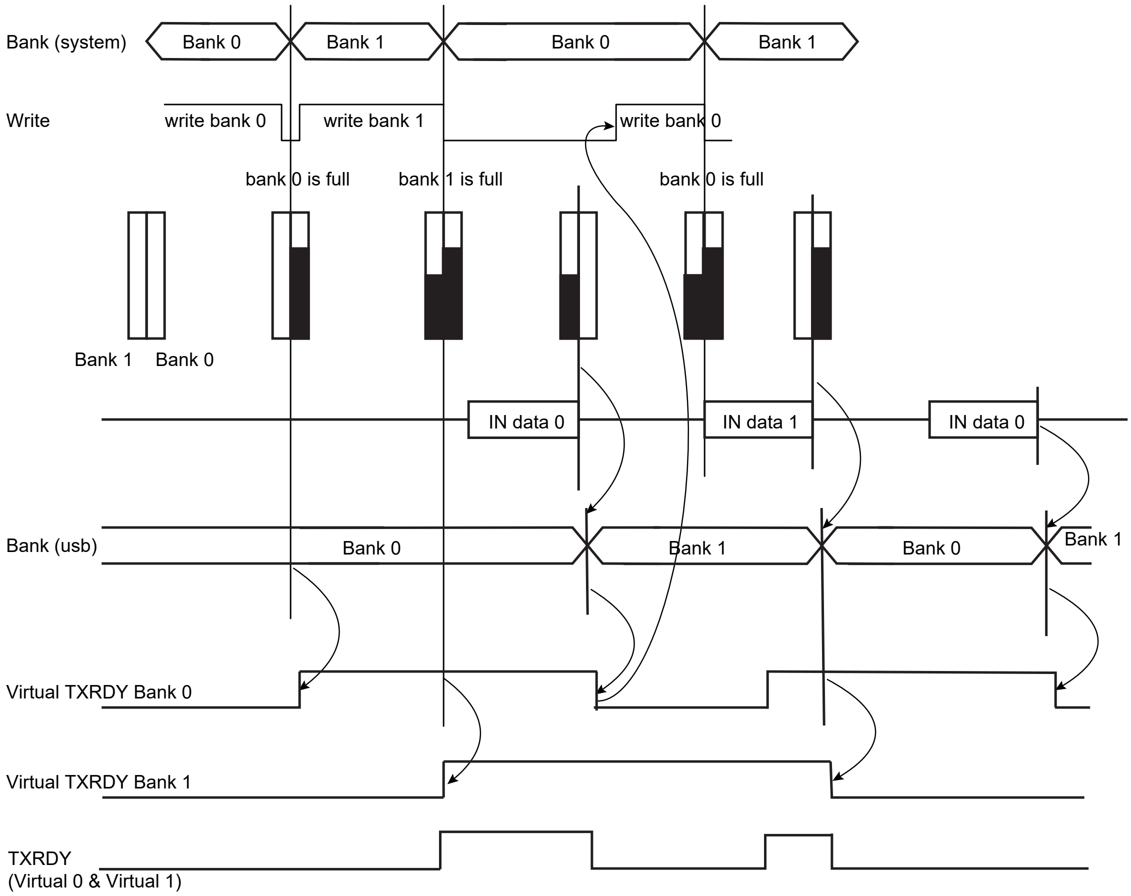
Note: A NAK handshake is always generated at the first status stage token.

Figure 72-9. Data OUT Followed by Status IN Transfer



Note: Before proceeding to the status stage, the software should determine that there is no risk of extra data from the host (data stage). If not certain (non-predictable data stage length), then the software should wait for a NAK-IN interrupt before proceeding to the status stage. This precaution should be taken to avoid collision in the FIFO.

Figure 72-10. Autovalid with DMA



Note: In the illustration above, Autovalid validates a bank as full, although this might not be the case, in order to continue processing data and to send to DMA.

- Isochronous IN
  Isochronous-IN is used to transmit a stream of data whose timing is implied by the delivery rate. Isochronous transfer provides periodic, continuous communication between host and device.

  It ensures bandwidth and low latencies appropriate for telephony, audio, video, etc.

  If the endpoint is not available (TXRDY_TRER = 0), then the device does not answer the host. An ERR_FL_ISO interrupt is generated in the UDPHS_EPTSTAx register and, once enabled, sent to the CPU.

  The STALL_SNT command bit is not used for an ISO-IN endpoint.
- High Bandwidth Isochronous Endpoint Handling: IN Example
  For high bandwidth isochronous endpoints, the DMA can be programmed with the number of transactions (UDPHS_DMACONTROLx.BUFF_LENGTH) and the system should provide the required number of packets per microframe, otherwise, the host will notice a sequencing problem.

  A response should be made to the first token IN recognized inside a microframe under the following conditions:
  - If at least one bank has been validated, the correct DATAx corresponding to the programmed Number Of Transactions per Microframe (NB_TRANS) should be answered. In case of a subsequent missed or corrupted token IN inside the microframe, the USB 2.0 Core available data bank(s) that should normally have been transmitted during that microframe shall be flushed at its end. If this flush occurs, an error condition is flagged (UDPHS_EPTSTAx.ERR_FLUSH is set).
  - If no bank is validated yet, the default DATA0 ZLP is answered and underflow is flagged (UDPHS_EPTSTAx.ERR_FL_ISO is set). Then, no data bank is flushed at the microframe end.
  - If no data bank has been validated at the time when a response should be made for the second transaction of NB_TRANS = 3 transactions microframe, a DATA1 ZLP is answered and underflow is flagged

(UDPHS_EPTSTAx.ERR_FL_ISO is set). If and only if remaining untransmitted banks for that microframe are available at its end, they are flushed and an error condition is flagged (UDPHS_EPTSTAx.ERR_FLUSH is set).
- If no data bank has been validated at the time when a response should be made for the last programmed transaction of a microframe, a DATA0 ZLP is answered and underflow is flagged (UDPHS_EPTSTAx.ERR_FL_ISO is set). If and only if the remaining untransmitted data bank for that microframe is available at its end, it is flushed and an error condition is flagged (UDPHS_EPTSTAx.ERR_FLUSH is set).

If at the end of a microframe no valid token IN has been recognized, no data bank is flushed and no error condition is reported.

At the end of a microframe in which at least one data bank has been transmitted, if less than NB_TRANS banks have been validated for that microframe, an error condition is flagged (UDPHS_EPTSTAx.ERR_TRANS is set).

Error cases (in UDPHS_EPTSTAx):

- ERR_FL_ISO: There was no data to transmit inside a microframe, so a ZLP is answered by default.
- ERR_FLUSH: At least one packet has been sent inside the microframe, but the number of token INs received is less than the number of transactions actually validated (TXRDY_TRER) and likewise with the NB_TRANS programmed.
- ERR_TRANS: At least one packet has been sent inside the microframe, but the number of token INs received is less than the number of programmed NB_TRANS transactions and the packets not requested were not validated.
- ERR_FL_ISO + ERR_FLUSH: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token INs.
- ERR_FL_ISO + ERR_TRANS: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token INs and the data can be discarded at the microframe end.
- ERR_FLUSH + ERR_TRANS: The first token IN has been answered and it was the only one received, a second bank has been validated but not the third, whereas NB_TRANS was waiting for three transactions.
- ERR_FL_ISO + ERR_FLUSH + ERR_TRANS: The first token IN has been treated, the data for the second Token IN was not available in time, but the second bank has been validated before the end of the microframe. The third bank has not been validated, but three transactions have been set in NB_TRANS.

72.6.9.4 Data OUT

- Bulk OUT or Interrupt OUT
  Like data IN, data OUT packets are sent by the host during the data or the status stage of control transfer or during an interrupt/bulk/isochronous OUT transfer. Data buffers are sent packet by packet under the control of the application or under the control of the DMA channel.
- Bulk OUT or Interrupt OUT: Receiving a Packet Under Application Control (Host to Device)
  Algorithm Description for Each Packet:
  - The application enables an interrupt on RXRDY_TXKL.
  - When an interrupt on RXRDY_TXKL is received, the application knows that UDPHS_EPTSTAx register BYTE_COUNT bytes have been received.
  - The application reads the BYTE_COUNT bytes from the endpoint.
  - The application clears RXRDY_TXKL.
    Note: If the application does not know the size of the transfer, it may not be a good option to use AUTO_VALID. Because if a zero-length-packet is received, the RXRDY_TXKL is automatically cleared by the AUTO_VALID hardware and if the endpoint interrupt is triggered, the software will not find its originating flag when reading the UDPHS_EPTSTAx register.

  Algorithm to Fill Several Packets

  - The application enables the interrupts of BUSY_BANK and AUTO_VALID.
  - When a BUSY_BANK interrupt is received, the application knows that all banks available for the endpoint have been filled. Thus, the application can read all banks available.

  If the application does not know the size of the receive buffer, instead of using the BUSY_BANK interrupt, the application must use RXRDY_TXKL.

- Bulk OUT or Interrupt OUT: Sending a Buffer Using DMA (Host To Device)

To use the DMA setting, the AUTO_VALID field is mandatory.

See Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host) for more information.

DMA Configuration Example:

1. First program UDPHS_DMAADDRESSx with the address of the buffer that should be transferred.
2. Enable the interrupt of the DMA in the Interrupt Enable register (UDPHS_IEN).
3. Program the DMA Channelx Control Register:
   – Size of buffer to be sent.
   – END_B_EN: Can be used for OUT packet truncation (discarding of unbuffered packet data) at the end of DMA buffer.
   – END_BUFFIT: Generate an interrupt when UDPHS_DMASTATUSx.BUFF_COUNT reaches 0.
   – END_TR_EN: End of transfer enable, the UDPHS device can put an end to the current DMA transfer, in case of a short packet.
   – END_TR_IT: End of transfer interrupt enable, an interrupt is sent after the last USB packet has been transferred by the DMA, if the USB transfer ended with a short packet. (Beneficial when the receive size is unknown.)
   – CHANN_ENB: Run and stop at end of buffer.

For OUT transfer, the bank will be automatically cleared by hardware when the application has read all the bytes in the bank (the bank is empty).

Notes:
1. When a zero-length-packet is received, UDPHS_EPTSTAx.RXRDY_TXKL is cleared automatically by AUTO_VALID, and the application knows of the end of buffer by the presence of the END_TR_IT.
2. If the host sends a zero-length packet, and the endpoint is free, then the device sends an ACK. No data is written in the endpoint, the RXRDY_TXKL interrupt is generated, and the UDPHS_EPTSTAx.BYTE_COUNT field is null.

Figure 72-11. Data OUT Transfer for Endpoint with One Bank

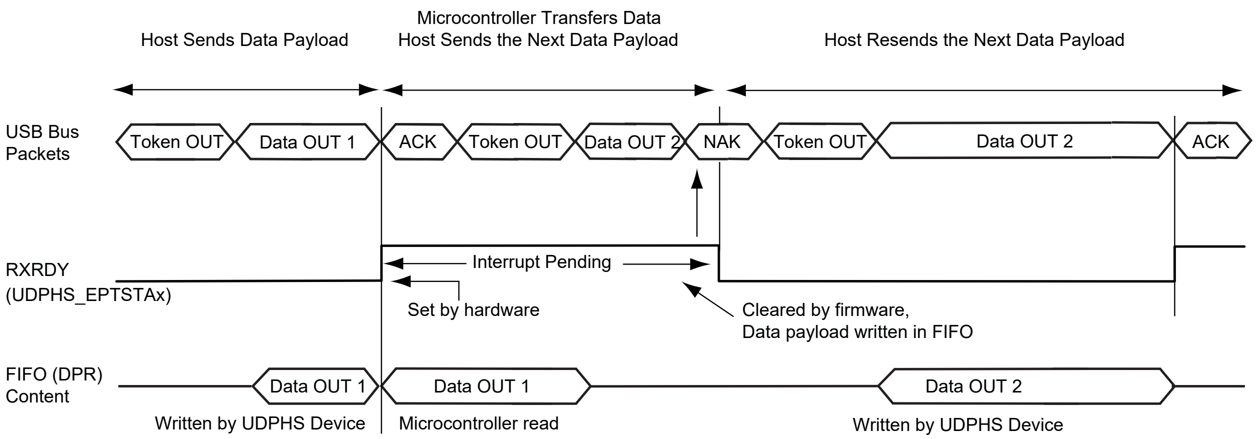
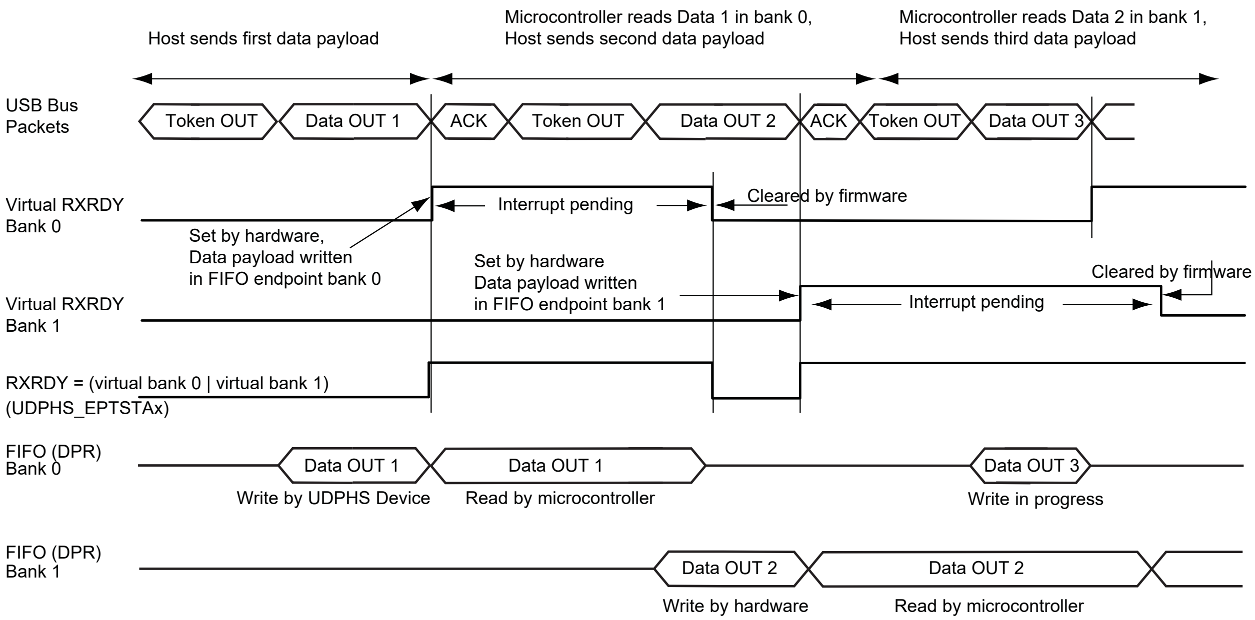


Figure 72-12. Data OUT Transfer for an Endpoint with Two Banks



- High Bandwidth Isochronous Endpoint OUT
  USB 2.0 supports individual High Speed isochronous endpoints that require data rates up to 192 Mb/s (24 MB/s): 3x1024 data bytes per microframe.

  To support such a rate, two or three banks may be used to buffer the three consecutive data packets. The microcontroller (or the DMA) should be able to empty the banks very rapidly (at least 24 MB/s on average).

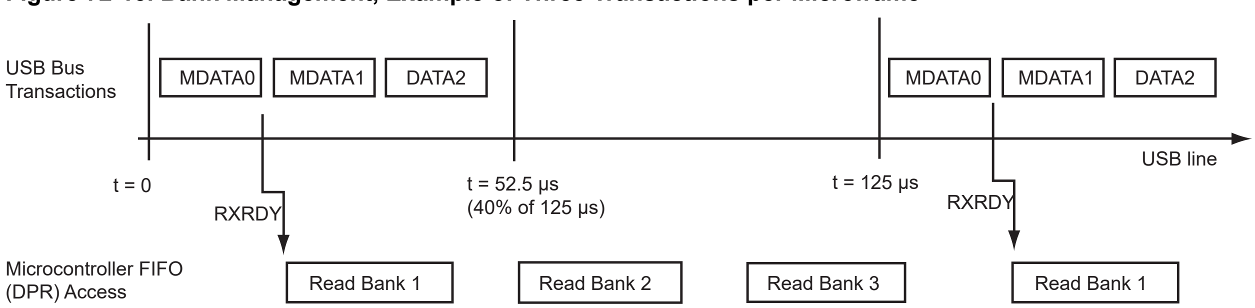
  NB_TRANS field in UDPHS_EPTCFGx register = Number Of Transactions per Microframe.

  If NB_TRANS > 1 then it is High Bandwidth.

  Example:

  – If NB_TRANS = 3, the sequence should be either one of the following:
    - MData0
    - MData0/Data1
    - MData0/Data1/Data2
  – If NB_TRANS = 2, the sequence should be either one of the following:
    - MData0
    - MData0/Data1
  – If NB_TRANS = 1, the sequence should be:
    - Data0

Figure 72-13. Bank Management, Example of Three Transactions per Microframe



- Isochronous Endpoint Handling: OUT Example

The user can ascertain the bank status (free or busy), and the toggle sequencing of the data packet for each bank with the UDPHS_EPTSTAx register in the three fields as follows:

– TOGGLESQ_STA: PID of the data stored in the current bank.
– CURBK: Number of the bank currently being accessed by the microcontroller.
– BUSY_BANK_STA: Number of busy bank.

This is particularly useful in case of a missing data packet.

If the inter-packet delay between the OUT token and the Data is greater than the USB standard, then the ISO-OUT transaction is ignored. (Payload data is not written, no interrupt is generated to the CPU.)

If there is a data CRC (Cyclic Redundancy Check) error, the payload is, none the less, written in the endpoint. The UDPHS_EPTSTAx.ERR_CRC_NTR flag is set.

If the endpoint is already full, the packet is not written in the DPRAM. The UDPHS_EPTSTAx.ERR_FL_ISO flag is set.

If the payload data is greater than the maximum size of the endpoint, then the ERR_OVFLW flag is set. It is the task of the CPU to manage this error. The data packet is written in the endpoint (except the extra data).

If the host sends a Zero Length Packet, and the endpoint is free, no data is written in the endpoint, the RXRDY_TXKL flag is set, and the UDPHS_EPTSTAx.BYTE_COUNT field is null.

The FRCESTALL command bit is unused for an isochronous endpoint.

Otherwise, payload data is written in the endpoint, the RXRDY_TXKL interrupt is generated and BYTE_COUNT is updated.

72.6.9.5 STALL

STALL is returned by a function in response to an IN token or after the data phase of an OUT or in response to a PING transaction. STALL indicates that a function is unable to transmit or receive data, or that a control pipe request is not supported.

- OUT

To stall an endpoint, set the FRCESTALL bit in UDPHS_EPTSETSTAx register and after the STALL_SNT flag has been set, set the TOGGLE_SEG bit in the UDPHS_EPTCLRSTAx register.

- IN

Set the FRCESTALL bit in UDPHS_EPTSETSTAx register.

Figure 72-14. Stall Handshake Data OUT Transfer

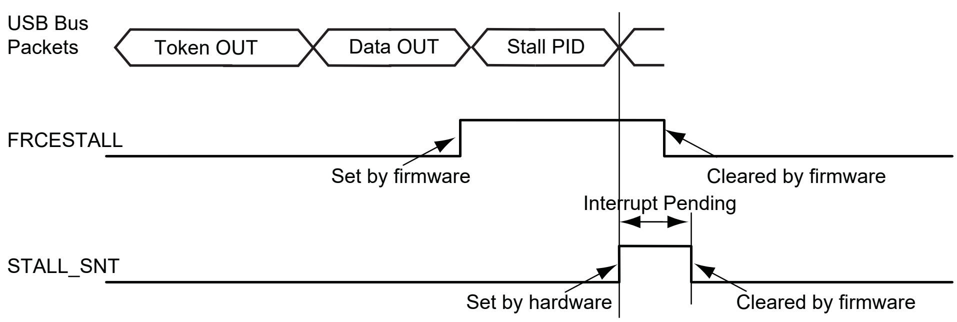
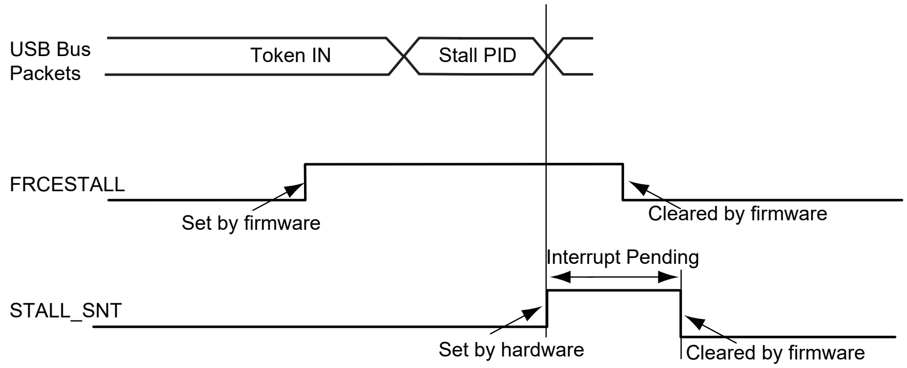


Figure 72-15. Stall Handshake Data IN Transfer



72.6.10 Speed Identification

The high speed reset is managed by hardware.

At the connection, the host makes a reset which could be a classic reset (full speed) or a high speed reset.

At the end of the reset process (full or high), the ENDRESET interrupt is generated.

Then the CPU should read the SPEED bit in UDPHS_INTSTAx to ascertain the speed mode of the device.

72.6.11 USB V2.0 High Speed Global Interrupt

Interrupts are defined in UDPHS_IEN and in UDPHS_INTSTA (Interrupt Status register).

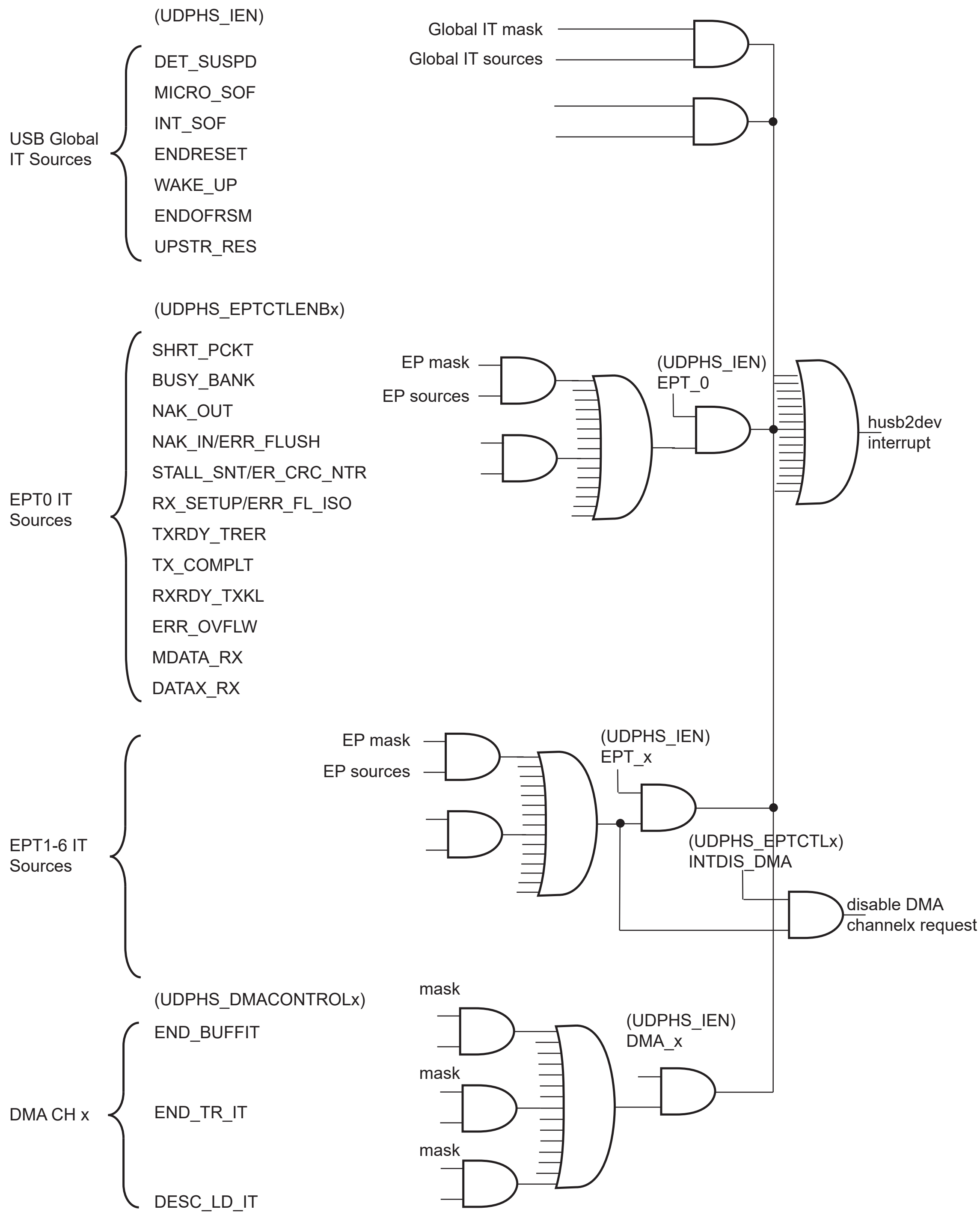
72.6.12 Endpoint Interrupts

Interrupts are enabled in UDPHS_IEN and individually masked in UDPHS_EPTCTLENBx.

Table 72-4. Endpoint Interrupt Source Masks

SHRT_PCKT	Short Packet Interrupt
BUSY_BANK	Busy Bank Interrupt
NAK_OUT	NAKOUT Interrupt
NAK_IN/ERR_FLUSH	NAKIN/Error Flush Interrupt
STALL_SNT/ERR_CRC_NTR	Stall Sent/CRC error/Number of Transaction Error Interrupt
RX_SETUP/ERR_FL_ISO	Received SETUP/Error Flow Interrupt
TXRDY_TRER	TX Packet Read/Transaction Error Interrupt
TX_COMPLT	Transmitted IN Data Complete Interrupt
RXRDY_TXKL	Received OUT Data Interrupt
ERR_OVFLW	Overflow Error Interrupt
MDATA_RX	MDATA Interrupt
DATAX_RX	DATAx Interrupt

Figure 72-16. UDPHS Interrupt Control Interface

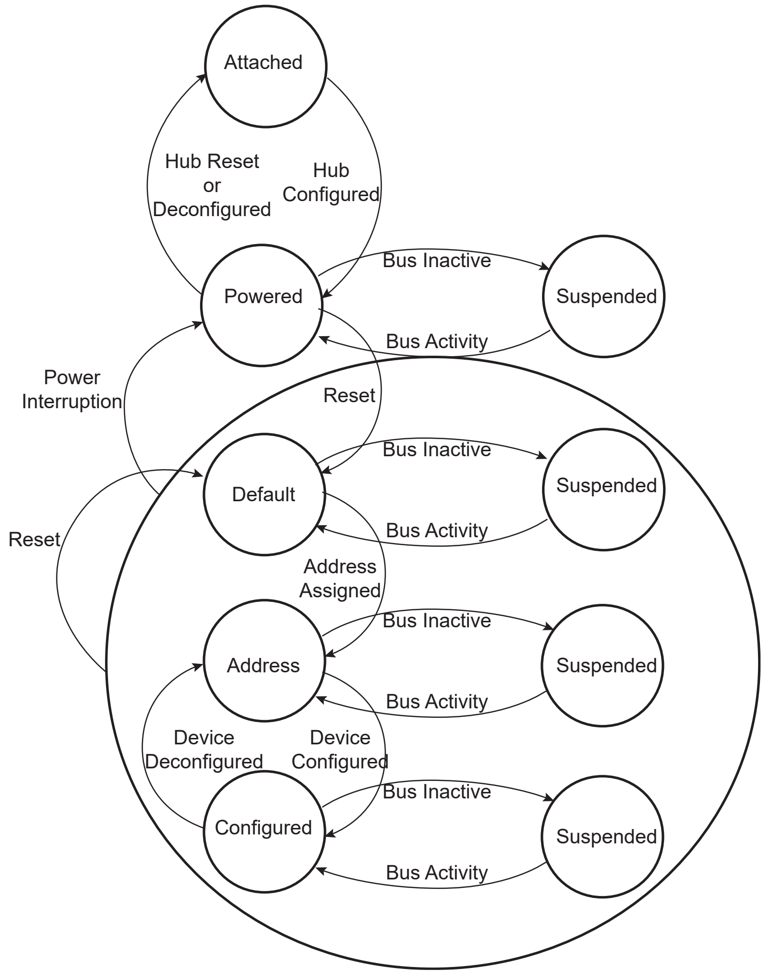


72.6.13 Power Modes

72.6.13.1 Controlling Device States

A USB device has several possible states. Refer to Chapter 9 (USB Device Framework) of the Universal Serial Bus Specification, Rev 2.0.

Figure 72-17. UDPHS Device State Diagram



Movement from one state to another depends on the USB bus state or on standard requests sent through control transactions via the default endpoint (endpoint 0).

After a period of bus inactivity, the USB device enters Suspend mode. Accepting Suspend/Resume requests from the USB host is mandatory. Constraints in Suspend mode are very strict for bus-powered applications; devices may not consume more than 500 µA on the USB bus.

While in Suspend mode, the host may wake up a device by sending a resume signal (bus activity) or a USB device may send a wakeup request to the host (waking up a PC by moving a USB mouse, for example).

The wakeup feature is not mandatory for all devices and must be negotiated with the host.

72.6.13.2 Not Powered State

Self powered devices can detect 5V VBUS using a PIO. When the device is not connected to a host, device power consumption can be reduced by the DETACH bit in UDPHS_CTRL. Disabling the transceiver is automatically done. HSDM, HSDP, FSDP and FSDP lines are tied to GND pulldowns integrated in the hub downstream ports.

72.6.13.3 Entering Attached State

When no device is connected, the USB FSDP and FSDM signals are tied to GND by 15 KΩ pulldowns integrated in the hub downstream ports. When a device is attached to an hub downstream port, the device connects a 1.5 KΩ pullup on FSDP. The USB bus line goes into IDLE state, FSDP is pulled up by the device 1.5 KΩ resistor to 3.3V and FSDM is pulled down by the 15 KΩ resistor to GND of the host.

After pullup connection, the device enters the powered state. The transceiver remains disabled until bus activity is detected.

In case of low power consumption need, the device can be stopped. When the device detects the VBUS, the software must enable the USB transceiver by enabling the EN_UDPHS bit in UDPHS_CTRL register.

The software can detach the pullup by setting DETACH bit in UDPHS_CTRL register.

72.6.13.4 From Powered State to Default State (Reset)

After its connection to a USB host, the USB device waits for an end-of-bus reset. The unmasked flag ENDRESET is set in the UDPHS_IEN register and an interrupt is triggered.

Once the ENDRESET interrupt has been triggered, the device enters Default State. In this state, the UDPHS software must:

- Enable the default endpoint, setting the EPT_ENABL flag in the UDPHS_EPTCTLENB[0] register and, optionally, enabling the interrupt for endpoint 0 by writing 1 in EPT_0 of the UDPHS_IEN register. The enumeration then begins by a control transfer.
- Configure the Interrupt Mask Register which has been reset by the USB reset detection
- Enable the transceiver.

In this state, the EN_UDPHS bit in UDPHS_CTRL register must be enabled.

72.6.13.5 From Default State to Address State (Address Assigned)

After a Set Address standard device request, the USB host peripheral enters the address state.

⚠WARNING Before the device enters address state, it must achieve the Status IN transaction of the control transfer, i.e., the UDPHS device sets its new address once the TX_COMPLT flag in the UDPHS_EPTCTL[0] register has been received and cleared.

To move to address state, the driver software sets the DEV_ADDR field and the FADDR_EN flag in the UDPHS_CTRL register.

72.6.13.6 From Address State to Configured State (Device Configured)

Once a valid Set Configuration standard request has been received and acknowledged, the device enables endpoints corresponding to the current configuration. This is done by setting the BK_NUMBER, EPT_TYPE, EPT_DIR and EPT_SIZE fields in the UDPHS_EPTCFGx registers and enabling them by setting the EPT_ENABL flag in the UDPHS_EPTCTLENBx registers, and, optionally, enabling corresponding interrupts in the UDPHS_IEN register.

72.6.13.7 Entering Suspend State (Bus Activity)

When a Suspend (no bus activity on the USB bus) is detected, the DET_SUSPD signal in the UDPHS_STA register is set. This triggers an interrupt if the corresponding bit is set in the UDPHS_IEN register. This flag is cleared by writing to the UDPHS_CLRINT register. Then the device enters Suspend mode.

In this state bus powered devices must drain less than 500 μA from the 5V VBUS. As an example, the microcontroller switches to slow clock, disables the PLL and main oscillator, and goes into Idle mode. It may also switch off other devices on the board.

The UDPHS device peripheral clocks can be switched off. Resume event is asynchronously detected.

72.6.13.8 Receiving a Host Resume

In Suspend mode, a resume event on the USB bus line is detected asynchronously, transceiver and clocks disabled (however, the pullup should not be removed).

Once the resume is detected on the bus, the signal WAKE_UP in the UDPHS_INTSTA is set. It may generate an interrupt if the corresponding bit in the UDPHS_IEN register is set. This interrupt may be used to wake up the core, enable PLL and main oscillators and configure clocks.

72.6.13.9 Sending an External Resume

In Suspend State it is possible to wake up the host by sending an external resume.

The device waits at least 5 ms after being entered in Suspend State before sending an external resume.

The device must force a K state from 1 to 15 ms to resume the host.

72.6.14 Test Mode

A device must support the TEST_MODE feature when in the Default, Address or Configured High Speed device states.

TEST_MODE can be:

- Test_J
- Test_K
- Test_Packet
- Test_SEO_NAK

(See UDPHS Test Register for definitions of each test mode.)

```
const char test_packet_buffer[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,                // JKJKJKJK * 9
0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,                     // JJKKJJKK * 8
0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,                     // JJKKJJKK * 8
0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF, // JJJJJJJKKKKKKK * 8
0x7F,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,                          // JJJJJJJK * 8
0xFC,0x7E,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,0x7E                 // {JKKKKKKK * 10}, JK
};
```

72.7 Register Summary

Notes: The registers below have two modes: Control, Bulk, Interrupt Endpoints mode and Isochronous Endpoints mode. In this register summary, both modes are displayed at the same offset.

- UDPHS_EPTCTLENB
- UDPHS_EPTCTLDIS
- UDPHS_EPTCTL
- UDPHS_EPTSETSTA
- UDPHS_EPTCLRSTA
- UDPHS_EPTSTA

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	UDPHS_CTRL	31:24								
		23:16								
		15:8					PULLD_DIS	REWAKEUP	DETACH	EN_UDPHS
		7:0	FADDR_EN	DEV_ADDR[6:0]						
0x04	UDPHS_FNUM	31:24	FNUM_ERR							
		23:16								
		15:8			FRAME_NUMBER[10:5]					
		7:0	FRAME_NUMBER[4:0]					MICRO_FRAME_NUM[2:0]		
0x08 ... 0x0F	Reserved									
0x10	UDPHS_IEN	31:24	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		15:8	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
0x14	UDPHS_INTSTA	31:24	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		15:8	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	SPEED
0x18	UDPHS_CLRINT	31:24								
		23:16								
		15:8								
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
0x1C	UDPHS_EPTRST	31:24								
		23:16								
		15:8	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		7:0	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
0x20 ... 0xDF	Reserved									
0xE0	UDPHS_TST	31:24								
		23:16								
		15:8								
		7:0			OPMODE2	TST_PKT	TST_K	TST_J	SPEED_CFG[1:0]	
0xE4 ... 0xFF	Reserved									
0x0100	UDPHS_EPTCFG0	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0104	UDPHS_EPTCTLENB0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0104	UDPHS_EPTCTLENB0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0108	UDPHS_EPTCTLDIS0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0108	UDPHS_EPTCTLDIS0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x010C	UDPHS_EPTCTL0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x010C	UDPHS_EPTCTL0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0110 ... 0x0113	Reserved									
0x0114	UDPHS_EPTSETSTA0	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0114	UDPHS_EPTSETSTA0	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0118	UDPHS_EPTCLRSTA0	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0118	UDPHS_EPTCLRSTA0	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x011C	UDPHS_EPTSTA0	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x011C	UDPHS_EPTSTA0	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0120	UDPHS_EPTCFG1	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0124	UDPHS_EPTCTLENB1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0124	UDPHS_EPTCTLENB1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0128	UDPHS_EPTCTLDIS1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0128	UDPHS_EPTCTLDIS1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x012C	UDPHS_EPTCTL1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x012C	UDPHS_EPTCTL1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0130 ... 0x0133	Reserved									
0x0134	UDPHS_EPTSETSTA1	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0134	UDPHS_EPTSETSTA1	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0138	UDPHS_EPTCLRSTA1	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0138	UDPHS_EPTCLRSTA1	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x013C	UDPHS_EPTSTA1	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x013C	UDPHS_EPTSTA1	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0140	UDPHS_EPTCFG2	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0144	UDPHS_EPTCTLENB2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0144	UDPHS_EPTCTLENB2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0148	UDPHS_EPTCTLDIS2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0148	UDPHS_EPTCTLDIS2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x014C	UDPHS_EPTCTL2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x014C	UDPHS_EPTCTL2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0150 ... 0x0153	Reserved									
0x0154	UDPHS_EPTSETSTA2	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0154	UDPHS_EPTSETSTA2	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0158	UDPHS_EPTCLRSTA2	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0158	UDPHS_EPTCLRSTA2	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x015C	UDPHS_EPTSTA2	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x015C	UDPHS_EPTSTA2	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0160	UDPHS_EPTCFG3	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0164	UDPHS_EPTCTLENB3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0164	UDPHS_EPTCTLENB3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0168	UDPHS_EPTCTLDIS3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0168	UDPHS_EPTCTLDIS3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x016C	UDPHS_EPTCTL3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x016C	UDPHS_EPTCTL3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0170 ... 0x0173	Reserved									
0x0174	UDPHS_EPTSETSTA3	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0174	UDPHS_EPTSETSTA3	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0178	UDPHS_EPTCLRSTA3	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0178	UDPHS_EPTCLRSTA3	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x017C	UDPHS_EPTSTA3	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x017C	UDPHS_EPTSTA3	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0180	UDPHS_EPTCFG4	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0184	UDPHS_EPTCTLENB4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0184	UDPHS_EPTCTLENB4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0188	UDPHS_EPTCTLDIS4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0188	UDPHS_EPTCTLDIS4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x018C	UDPHS_EPTCTL4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x018C	UDPHS_EPTCTL4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0190 ... 0x0193	Reserved									
0x0194	UDPHS_EPTSETSTA4	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0194	UDPHS_EPTSETSTA4	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0198	UDPHS_EPTCLRSTA4	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0198	UDPHS_EPTCLRSTA4	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x019C	UDPHS_EPTSTA4	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x019C	UDPHS_EPTSTA4	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x01A0	UDPHS_EPTCFG5	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x01A4	UDPHS_EPTCTLENB5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01A4	UDPHS_EPTCTLENB5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01A8	UDPHS_EPTCTLDIS5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01A8	UDPHS_EPTCTLDIS5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01AC	UDPHS_EPTCTL5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01AC	UDPHS_EPTCTL5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01B0 ... 0x01B3	Reserved									
0x01B4	UDPHS_EPTSETSTA5	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01B4	UDPHS_EPTSETSTA5	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x01B8	UDPHS_EPTCLRSTA5	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x01B8	UDPHS_EPTCLRSTA5	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x01BC	UDPHS_EPTSTA5	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x01BC	UDPHS_EPTSTA5	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x01C0	UDPHS_EPTCFG6	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x01C4	UDPHS_EPTCTLENB6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01C4	UDPHS_EPTCTLENB6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01C8	UDPHS_EPTCTLDIS6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01C8	UDPHS_EPTCTLDIS6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01CC	UDPHS_EPTCTL6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01CC	UDPHS_EPTCTL6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01D0 ... 0x01D3	Reserved									
0x01D4	UDPHS_EPTSETSTA6	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x01D4	UDPHS_EPTSETSTA6	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x01D8	UDPHS_EPTCLRSTA6	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x01D8	UDPHS_EPTCLRSTA6	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x01DC	UDPHS_EPTSTA6	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x01DC	UDPHS_EPTSTA6	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x01E0	UDPHS_EPTCFG7	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x01E4	UDPHS_EPTCTLENB7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01E4	UDPHS_EPTCTLENB7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01E8	UDPHS_EPTCTLDIS7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01E8	UDPHS_EPTCTLDIS7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01EC	UDPHS_EPTCTL7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01EC	UDPHS_EPTCTL7	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01F0 ... 0x01F3	Reserved									
0x01F4	UDPHS_EPTSETSTA7	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x01F4	UDPHS_EPTSETSTA7	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x01F8	UDPHS_EPTCLRSTA7	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x01F8	UDPHS_EPTCLRSTA7	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x01FC	UDPHS_EPTSTA7	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x01FC	UDPHS_EPTSTA7	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0200	UDPHS_EPTCFG8	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR		EPT_SIZE[2:0]	
0x0204	UDPHS_EPTCTLENB8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0204	UDPHS_EPTCTLENB8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0208	UDPHS_EPTCTLDIS8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0208	UDPHS_EPTCTLDIS8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x020C	UDPHS_EPTCTL8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x020C	UDPHS_EPTCTL8	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0210 ... 0x0213	Reserved									
0x0214	UDPHS_EPTSETSTA8	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0214	UDPHS_EPTSETSTA8	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0218	UDPHS_EPTCLRSTA8	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					

...........continued										
Offset	**Name**	**Bit Pos.**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0218	UDPHS_EPTCLRSTA8	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x021C	UDPHS_EPTSTA8	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x021C	UDPHS_EPTSTA8	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0220	UDPHS_EPTCFG9	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0224	UDPHS_EPTCTLENB9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0224	UDPHS_EPTCTLENB9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0228	UDPHS_EPTCTLDIS9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0228	UDPHS_EPTCTLDIS9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x022C	UDPHS_EPTCTL9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x022C	UDPHS_EPTCTL9	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0230 ... 0x0233	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0234	UDPHS_EPTSETSTA9	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0234	UDPHS_EPTSETSTA9	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0238	UDPHS_EPTCLRSTA9	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0238	UDPHS_EPTCLRSTA9	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x023C	UDPHS_EPTSTA9	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x023C	UDPHS_EPTSTA9	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0240	UDPHS_EPTCFG10	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0244	UDPHS_EPTCTLENB10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0244	UDPHS_EPTCTLENB10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0248	UDPHS_EPTCTLDIS10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0248	UDPHS_EPTCTLDIS10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x024C	UDPHS_EPTCTL10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x024C	UDPHS_EPTCTL10	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0250 ... 0x0253	Reserved									
0x0254	UDPHS_EPTSETSTA10	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0254	UDPHS_EPTSETSTA10	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0258	UDPHS_EPTCLRSTA10	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0258	UDPHS_EPTCLRSTA10	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x025C	UDPHS_EPTSTA10	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x025C	UDPHS_EPTSTA10	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0260	UDPHS_EPTCFG11	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0264	UDPHS_EPTCTLENB11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0264	UDPHS_EPTCTLENB11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0268	UDPHS_EPTCTLDIS11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0268	UDPHS_EPTCTLDIS11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x026C	UDPHS_EPTCTL11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x026C	UDPHS_EPTCTL11	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0270 ... 0x0273	Reserved									
0x0274	UDPHS_EPTSETSTA11	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0274	UDPHS_EPTSETSTA11	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0278	UDPHS_EPTCLRSTA11	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0278	UDPHS_EPTCLRSTA11	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x027C	UDPHS_EPTSTA11	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x027C	UDPHS_EPTSTA11	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0280	UDPHS_EPTCFG12	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0284	UDPHS_EPTCTLENB12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0284	UDPHS_EPTCTLENB12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0288	UDPHS_EPTCTLDIS12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0288	UDPHS_EPTCTLDIS12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x028C	UDPHS_EPTCTL12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x028C	UDPHS_EPTCTL12	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0290 ... 0x0293	Reserved									
0x0294	UDPHS_EPTSETSTA12	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0294	UDPHS_EPTSETSTA12	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0298	UDPHS_EPTCLRSTA12	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0298	UDPHS_EPTCLRSTA12	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x029C	UDPHS_EPTSTA12	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x029C	UDPHS_EPTSTA12	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x02A0	UDPHS_EPTCFG13	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x02A4	UDPHS_EPTCTLENB13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02A4	UDPHS_EPTCTLENB13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02A8	UDPHS_EPTCTLDIS13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x02A8	UDPHS_EPTCTLDIS13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x02AC	UDPHS_EPTCTL13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02AC	UDPHS_EPTCTL13	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02B0 ... 0x02B3	Reserved									
0x02B4	UDPHS_EPTSETSTA13	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x02B4	UDPHS_EPTSETSTA13	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x02B8	UDPHS_EPTCLRSTA13	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x02B8	UDPHS_EPTCLRSTA13	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x02BC	UDPHS_EPTSTA13	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x02BC	UDPHS_EPTSTA13	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x02C0	UDPHS_EPTCFG14	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x02C4	UDPHS_EPTCTLENB14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02C4	UDPHS_EPTCTLENB14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02C8	UDPHS_EPTCTLDIS14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02C8	UDPHS_EPTCTLDIS14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x02CC	UDPHS_EPTCTL14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02CC	UDPHS_EPTCTL14	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02D0 ... 0x02D3	Reserved									
0x02D4	UDPHS_EPTSETSTA14	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x02D4	UDPHS_EPTSETSTA14	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x02D8	UDPHS_EPTCLRSTA14	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x02D8	UDPHS_EPTCLRSTA14	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x02DC	UDPHS_EPTSTA14	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x02DC	UDPHS_EPTSTA14	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x02E0	UDPHS_EPTCFG15	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02E4	UDPHS_EPTCTLENB15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02E4	UDPHS_EPTCTLENB15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02E8	UDPHS_EPTCTLDIS15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x02E8	UDPHS_EPTCTLDIS15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x02EC	UDPHS_EPTCTL15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02EC	UDPHS_EPTCTL15	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x02F0 ... 0x02F3	Reserved									
0x02F4	UDPHS_EPTSETSTA15	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x02F4	UDPHS_EPTSETSTA15	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x02F8	UDPHS_EPTCLRSTA15	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x02F8	UDPHS_EPTCLRSTA15	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02FC	UDPHS_EPTSTA15	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x02FC	UDPHS_EPTSTA15	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0300 ... 0x030F	Reserved									
0x0310	UDPHS_DMANXTDSC1	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0314	UDPHS_DMAADDRESS1	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0318	UDPHS_DMACONTROL1	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x031C	UDPHS_DMASTATUS1	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0320	UDPHS_DMANXTDSC2	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0324	UDPHS_DMAADDRESS2	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0328	UDPHS_DMACONTROL2	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x032C	UDPHS_DMASTATUS2	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0330	UDPHS_DMANXTDSC3	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0334	UDPHS_DMAADDRESS3	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0338	UDPHS_DMACONTROL3	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x033C	UDPHS_DMASTATUS3	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0340	UDPHS_DMANXTDSC4	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0344	UDPHS_DMAADDRESS4	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0348	UDPHS_DMACONTROL4	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x034C	UDPHS_DMASTATUS4	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0350	UDPHS_DMANXTDSC5	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0354	UDPHS_DMAADDRESS5	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0358	UDPHS_DMACONTROL5	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x035C	UDPHS_DMASTATUS5	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0360	UDPHS_DMANXTDSC6	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0364	UDPHS_DMAADDRESS6	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0368	UDPHS_DMACONTROL6	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x036C	UDPHS_DMASTATUS6	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0370	UDPHS_DMANXTDSC7	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0374	UDPHS_DMAADDRESS7	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0378	UDPHS_DMACONTROL7	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x037C	UDPHS_DMASTATUS7	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB

72.7.1 UDPHS Control Register

Name: UDPHS_CTRL
Offset: 0x00
Reset: 0x00000200
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					PULLD_DIS	REWAKEUP	DETACH	EN_UDPHS
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit	7	6	5	4	3	2	1	0
	FADDR_EN	DEV_ADDR[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – PULLD_DIS Pulldown Disable (cleared upon USB reset)
When set, there is no pulldown on DP & DM. (DM Pulldown = DP Pulldown = 0).
Note: If the DETACH bit is also set, device DP & DM are left in High Impedance state.

See description of bit “DETACH”.

DETACH	PULLD_DIS	DP	DM	Condition
0	0	Pullup	Pulldown	Not recommended
0	1	Pullup	High impedance state	VBUS present
1	0	Pulldown	Pulldown	No VBUS
1	1	High Impedance state	High Impedance state	VBUS present & software disconnect

Bit 10 – REWAKEUP Send Remote Wakeup (cleared upon USB reset)
An Upstream Resume is sent only after the UDPHS bus has been in Suspend state for at least 5 ms.
This bit is automatically cleared by hardware at the end of the Upstream Resume.

Value	Description
0	Remote Wakeup is disabled (read), or this bit has no effect (write).
1	Remote Wakeup is enabled (read), or this bit forces an external interrupt on the UDPHS controller for Remote Wakeup purposes.

Bit 9 – DETACH Detach Command
See description of bit “PULL_DIS”.

Value	Description
0	UDPHS is attached (read), or this bit pulls up the DP line (attach command) (write).
1	UDPHS is detached, UTMI transceiver is suspended (read), or this bit simulates a detach on the UDPHS line and forces the UTMI transceiver into Suspend state (Suspend M = 0) (write).

Bit 8 – EN_UDPHS UDPHS Enable

Value	Description
0	UDPHS is disabled (read), or this bit disables and resets the UDPHS controller (write). Switch the UHPHS to UTMI.
1	UDPHS is enabled (read), or this bit enables the UDPHS controller (write). Switch the UDPHS to UTMI.

Bit 7 – FADDR_EN Function Address Enable (cleared upon USB reset)

Value	Description
0	The device is not in Address state (read), or only the default function address is used (write).
1	The device is in Address state (read), or this bit is set by the device firmware after a successful status phase of a SET_ADDRESS transaction (write). When set, the only address accepted by the UDPHS controller is the one stored in the UDPHS Address field. It will not be cleared afterwards by the device firmware. It is cleared by hardware on hardware reset, or when UDPHS bus reset is received.

Bits 6:0 – DEV_ADDR[6:0] UDPHS Address (cleared upon USB reset)
This field contains the default address (0) after power-up or UDPHS bus reset (read), or it is written with the value set by a SET_ADDRESS request received by the device firmware (write).

72.7.2 UDPHS Frame Number Register

Name: UDPHS_FNUM
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FNUM_ERR							
Access	R							
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			FRAME_NUMBER[10:5]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FRAME_NUMBER[4:0]					MICRO_FRAME_NUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – FNUM_ERR Frame Number CRC Error (cleared upon USB reset)
This bit is set by hardware when a corrupted Frame Number in Start of Frame packet (or Micro SOF) is received.
This bit and the INT_SOF (or MICRO_SOF) interrupt are updated at the same time.

Bits 13:3 – FRAME_NUMBER[10:0] Frame Number as defined in the Packet Field Formats (cleared upon USB reset)
This field is provided in the last received SOF packet (see UDPHS_IEN.INT_SOF).

Bits 2:0 – MICRO_FRAME_NUM[2:0] Microframe Number (cleared upon USB reset)
Number of the received microframe (0 to 7) in one frame.This field is reset at the beginning of each new frame (1 ms).
One microframe is received each 125 microseconds (1 ms/8).

72.7.3 UDPHS Interrupt Enable Register

Name: UDPHS_IEN
Offset: 0x10
Reset: 0x00000010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	1	0	0	0	

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_x DMA Channel x Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable the interrupts for this channel.
1	Enable the interrupts for this channel.

Bits 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 – EPT_x Endpoint x Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable the interrupts for this endpoint.
1	Enable the interrupts for this endpoint.

Bit 7 – UPSTR_RES Upstream Resume Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Upstream Resume Interrupt.
1	Enable Upstream Resume Interrupt.

Bit 6 – ENDOFRSM End Of Resume Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Resume Interrupt.
1	Enable Resume Interrupt.

Bit 5 – WAKE_UP Wake Up CPU Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Wake-up CPU Interrupt.
1	Enable Wake-up CPU Interrupt.

Bit 4 – ENDRESET End Of Reset Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable End Of Reset Interrupt.
1	Enable End Of Reset Interrupt. Automatically enabled after USB reset.

Bit 3 – INT_SOF SOF Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable SOF Interrupt.
1	Enable SOF Interrupt.

Bit 2 – MICRO_SOF Micro-SOF Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Micro-SOF Interrupt.
1	Enable Micro-SOF Interrupt.

Bit 1 – DET_SUSPD Suspend Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Suspend Interrupt.
1	Enable Suspend Interrupt.

72.7.4 UDPHS Interrupt Status Register

Name: UDPHS_INTSTA
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	SPEED
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_x DMA Channel x Interrupt

Value	Description
0	Reset when the UDPHS_DMASTATUSx interrupt source is cleared.
1	Set by hardware when an interrupt is triggered by the DMA Channelx and this endpoint interrupt is enabled by the DMA_x bit in UDPHS_IEN.

Bits 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 – EPT_x Endpoint x Interrupt (cleared upon USB reset)

Value	Description
0	Reset when the UDPHS_EPTSTAx interrupt source is cleared.
1	Set by hardware when an interrupt is triggered by the UDPHS_EPTSTAx register and this endpoint interrupt is enabled by the EPT_x bit in UDPHS_IEN.

Bit 7 – UPSTR_RES Upstream Resume Interrupt

Value	Description
0	Cleared by setting the UPSTR_RES bit in UDPHS_CLRINT.
1	Set by hardware when the UDPHS controller is sending a resume signal called “upstream resume”. This triggers a UDPHS interrupt when the UPSTR_RES bit is set in UDPHS_IEN.

Bit 6 – ENDOFRSM End Of Resume Interrupt

Value	Description
0	Cleared by setting the ENDOFRSM bit in UDPHS_CLRINT.
1	Set by hardware when the UDPHS controller detects a good end of resume signal initiated by the host. This triggers a UDPHS interrupt when the ENDOFRSM bit is set in UDPHS_IEN.

Bit 5 – WAKE_UP Wake Up CPU Interrupt

Value	Description
0	Cleared by setting the WAKE_UP bit in UDPHS_CLRINT.

Value	Description
1	Set by hardware when the UDPHS controller is in SUSPEND state and is re-activated by a filtered non-idle signal from the UDPHS line (not by an upstream resume). This triggers a UDPHS interrupt when the WAKE_UP bit is set in UDPHS_IEN register. When receiving this interrupt, the user has to enable the device controller clock prior to operation. **Note:** this interrupt is generated even if the device controller clock is disabled.

Bit 4 – ENDRESET End Of Reset Interrupt

Value	Description
0	Cleared by setting the ENDRESET bit in UDPHS_CLRINT.
1	Set by hardware when an End Of Reset has been detected by the UDPHS controller. This triggers a UDPHS interrupt when the ENDRESET bit is set in UDPHS_IEN.

Bit 3 – INT_SOF Start Of Frame Interrupt

Note: The Micro Start Of Frame Interrupt (MICRO_SOF), and the Start Of Frame Interrupt (INT_SOF) are not generated at the same time.

Value	Description
0	Cleared by setting the INT_SOF bit in UDPHS_CLRINT.
1	Set by hardware when an UDPHS Start Of Frame PID (SOF) has been detected (every 1 ms) or synthesized by the macro. This triggers a UDPHS interrupt when the INT_SOF bit is set in UDPHS_IEN register. In case of detected SOF, in High Speed mode, the MICRO_FRAME_NUMBER field is cleared in UDPHS_FNUM register and the FRAME_NUMBER field is updated.

Bit 2 – MICRO_SOF Micro Start Of Frame Interrupt

Note: The Micro Start Of Frame Interrupt (MICRO_SOF), and the Start Of Frame Interrupt (INT_SOF) are not generated at the same time.

Value	Description
0	Cleared by setting the MICRO_SOF bit in UDPHS_CLRINT register.
1	Set by hardware when an UDPHS micro start of frame PID (SOF) has been detected (every 125 us) or synthesized by the macro. This triggers a UDPHS interrupt when the MICRO_SOF bit is set in UDPHS_IEN. In case of detected SOF, the MICRO_FRAME_NUM field in UDPHS_FNUM register is incremented and the FRAME_NUMBER field does not change.

Bit 1 – DET_SUSPD Suspend Interrupt

Value	Description
0	Cleared by setting the DET_SUSPD bit in UDPHS_CLRINT register.
1	Set by hardware when a UDPHS Suspend (Idle bus for three frame periods, a J state for 3 ms) is detected. This triggers a UDPHS interrupt when the DET_SUSPD bit is set in UDPHS_IEN register.

Bit 0 – SPEED Speed Status

Value	Description
0	Reset by hardware when the hardware is in Full Speed mode.
1	Set by hardware when the hardware is in High Speed mode.

72.7.5 UDPHS Clear Interrupt Register

Name: UDPHS_CLRINT
Offset: 0x18
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit 7 – UPSTR_RES Upstream Resume Interrupt Clear

Value	Description
0	No effect.
1	Clear the UPSTR_RES bit in UDPHS_INTSTA.

Bit 6 – ENDOFRSM End Of Resume Interrupt Clear

Value	Description
0	No effect.
1	Clear the ENDOFRSM bit in UDPHS_INTSTA.

Bit 5 – WAKE_UP Wake Up CPU Interrupt Clear

Value	Description
0	No effect.
1	Clear the WAKE_UP bit in UDPHS_INTSTA.

Bit 4 – ENDRESET End Of Reset Interrupt Clear

Value	Description
0	No effect.
1	Clear the ENDRESET bit in UDPHS_INTSTA.

Bit 3 – INT_SOF Start Of Frame Interrupt Clear

Value	Description
0	No effect.
1	Clear the INT_SOF bit in UDPHS_INTSTA.

Bit 2 – MICRO_SOF Micro Start Of Frame Interrupt Clear

Value	Description
0	No effect.
1	Clear the MICRO_SOF bit in UDPHS_INTSTA.

Bit 1 – DET_SUSPD Suspend Interrupt Clear

Value	Description
0	No effect.
1	Clear the DET_SUSPD bit in UDPHS_INTSTA.

72.7.6 UDPHS Endpoints Reset Register

Name: UDPHS_EPTRST
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Access								
Reset								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EPT_x Endpoint x Reset
Setting this bit clears all bits in Endpoint Status register (UDPHS_EPTSTAx), except the TOGGLESQ_STA field.

Value	Description
0	No effect.
1	Reset the Endpointx state.

72.7.7 UDPHS Test Register

Name: UDPHS_TST
Offset: 0xE0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			OPMODE2	TST_PKT	TST_K	TST_J	SPEED_CFG[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – OPMODE2 OpMode2
Note: For the Test mode, Test_SE0_NAK (refer to Universal Serial Bus Specification, Revision 2.0: 7.1.20, Test Mode Support). Force the device in High Speed mode, and configure a bulk-type endpoint. Do not fill this endpoint for sending NAK to the host.
Upon command, a port's transceiver must enter the High Speed Receive mode and remain in that mode until the exit action is taken. This enables the testing of output impedance, low level output voltage and loading characteristics. In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal allowed device response time. This enables testing of the device squelch level circuitry and, additionally, provides a general purpose stimulus/response test for basic functional testing.

Value	Description
0	No effect.
1	Set to force the OpMode signal (UTMI interface) to “10”, to disable the bit-stuffing and the NRZI encoding.

Bit 4 – TST_PKT Test Packet Mode

Value	Description
0	No effect.
1	Set to repetitively transmit the packet stored in the current bank. This enables the testing of rise and fall times, eye patterns, jitter, and any other dynamic waveform specifications.

Bit 3 – TST_K Test K Mode

Value	Description
0	No effect.
1	Set to send the K state on the UDPHS line. This enables the testing of the high output drive level on the D- line.

Bit 2 – TST_J Test J Mode

Value	Description
0	No effect.
1	Set to send the J state on the UDPHS line. This enables the testing of the high output drive level on the D+ line.

Bits 1:0 – SPEED_CFG[1:0] Speed Configuration

Value	Name	Description
0	NORMAL	Normal mode: The macro is in Full Speed mode, ready to make a High Speed identification, if the host supports it and then to automatically switch to High Speed mode.
1	–	Reserved
2	HIGH_SPEED	Force High Speed: Set this value to force the hardware to work in High Speed mode. Only for debug or test purpose.
3	FULL_SPEED	Force Full Speed: Set this value to force the hardware to work only in Full Speed mode. In this configuration, the macro will not respond to a High Speed reset handshake.

72.7.8 UDPHS Endpoint Configuration Register

Name: UDPHS_EPTCFGx
Offset: 0x0100 + x*0x20 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EPT_MAPD							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							NB_TRANS[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – EPT_MAPD Endpoint Mapped (cleared upon USB reset)

Value	Description
0	The user should reprogram the register with correct values.
1	Set by hardware when the endpoint size (EPT_SIZE) and the number of banks (BK_NUMBER) are correct regarding: – The max endpoint size for this endpoint – The number of allowed banks for this endpoint

Bits 9:8 – NB_TRANS[1:0] Number Of Transactions per Microframe (cleared upon USB reset)
The number of transactions per microframe is set by software.
Note: Meaningful for high bandwidth isochronous endpoint only.

Bits 7:6 – BK_NUMBER[1:0] Number of Banks (cleared upon USB reset)
Set this field according to the endpoint's number of banks (see Endpoint Configuration).

Value	Name	Description
0	0	Zero bank, the endpoint is not mapped in memory
1	1	One bank (bank 0)
2	2	Double bank (Ping-Pong: bank0/bank1)
3	3	Triple bank (bank0/bank1/bank2)

Bits 5:4 – EPT_TYPE[1:0] Endpoint Type (cleared upon USB reset)
Set this field according to the endpoint type (see Endpoint Configuration).
(Endpoint 0 should always be configured as control).

Value	Name	Description
0	CTRL8	Control endpoint
1	ISO	Isochronous endpoint
2	BULK	Bulk endpoint
3	INT	Interrupt endpoint

Bit 3 – EPT_DIR Endpoint Direction (cleared upon USB reset)
For Control endpoints this bit has no effect and should be left at zero.

Value	Description
0	Clear this bit to configure OUT direction for Bulk, Interrupt and Isochronous endpoints.
1	Set this bit to configure IN direction for Bulk, Interrupt and Isochronous endpoints.

Bits 2:0 – EPT_SIZE[2:0] Endpoint Size (cleared upon USB reset)
Set this field according to the endpoint size in bytes (see Endpoint Configuration). Note that 1024 bytes is only for isochronous endpoints.

Value	Name	Description
0	8	8 bytes
1	16	16 bytes
2	32	32 bytes
3	64	64 bytes
4	128	128 bytes
5	256	256 bytes
6	512	512 bytes
7	1024	1024 bytes

72.7.9 UDPHS Endpoint Control Enable Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLENBx
Offset: 0x0104 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

For additional information, see UDPHS_EPTCTLx.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access				W	W		W	W
Reset				–	–		–	–

Bit 31 – SHRT_PCKT Short Packet Send/Short Packet Interrupt Enable
For IN endpoints: Ensures short packet at end of DMA Transfer if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTOVALID bits are also set.
For OUT endpoints:

Value	Description
0	No effect.
1	Enable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enable

Value	Description
0	No effect.
1	Enable Busy Bank Interrupt.

Bit 15 – NAK_OUT NAKOUT Interrupt Enable

Value	Description
0	No effect.
1	Enable NAKOUT Interrupt.

Bit 14 – NAK_IN NAKIN Interrupt Enable

Value	Description
0	No effect.
1	Enable NAKIN Interrupt.

Bit 13 – STALL_SNT Stall Sent Interrupt Enable

Value	Description
0	No effect.

Value	Description
1	Enable Stall Sent Interrupt.

Bit 12 – RX_SETUP Received SETUP

Value	Description
0	No effect.
1	Enable RX_SETUP Interrupt.

Bit 11 – TXRDY TX Packet Ready Interrupt Enable

Value	Description
0	No effect.
1	Enable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enable

Value	Description
0	No effect.
1	Enable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enable

Value	Description
0	No effect.
1	Enable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Overflow Error Interrupt.

Bit 4 – NYET_DIS NYET Disable (Only for High Speed Bulk OUT endpoints)

Value	Description
0	No effect.
1	Forces an ACK response to the next High Speed Bulk OUT transfer instead of a NYET response.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled.

Bit 1 – AUTO_VALID Packet Auto-Valid Enable

Value	Description
0	No effect.
1	Enable this bit to automatically validate the current packet and switch to the next bank for both IN and OUT transfers.

Bit 0 – EPT_ENABL Endpoint Enable

Value	Description
0	No effect.
1	Enable endpoint according to the device configuration.

72.7.10 UDPHS Endpoint Control Enable Register (Isochronous Endpoints)

Name: UDPHS_EPTCTLENBx
Offset: 0x0104 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

For additional information, see UDPHS_EPTCTLx.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access	W	W			W		W	W
Reset	–	–			–		–	–

Bit 31 – SHRT_PCKT Short Packet Send/Short Packet Interrupt Enable
For IN endpoints: Ensures short packet at end of DMA Transfer if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTOVALID bits are also set.
For OUT endpoints:

Value	Description
0	No effect.
1	Enable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enable

Value	Description
0	No effect.
1	Enable Busy Bank Interrupt.

Bit 14 – ERR_FLUSH Bank Flush Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Bank Flush Error Interrupt.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Error CRC ISO/Error Number of Transaction Interrupt.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Enable

Value	Description
0	No effect.
1	Enable Error Flow ISO Interrupt.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Enable

Value	Description
0	No effect.
1	Enable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enable

Value	Description
0	No effect.
1	Enable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enable

Value	Description
0	No effect.
1	Enable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Overflow Error Interrupt.

Bit 7 – MDATA_RX MDATA Interrupt Enable (Only for high bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Enable MDATA Interrupt.

Bit 6 – DATAX_RX DATAx Interrupt Enable (Only for high bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Enable DATAx Interrupt.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled.

Bit 1 – AUTO_VALID Packet Auto-Valid Enable

Value	Description
0	No effect.
1	Enable this bit to automatically validate the current packet and switch to the next bank for both IN and OUT transfers.

Bit 0 – EPT_ENABL Endpoint Enable

Value	Description
0	No effect.
1	Enable endpoint according to the device configuration.

72.7.11 UDPHS Endpoint Control Disable Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLDISx
Offset: 0x0108 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

For additional information, see UDPHS_EPTCTLx.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
Access				W	W		W	W
Reset				–	–		–	–

Bit 31 – SHRT_PCKT Short Packet Interrupt Disable
For IN endpoints: Never automatically add a zero length packet at end of DMA transfer.
For OUT endpoints:

Value	Description
0	No effect.
1	Disable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Disable

Value	Description
0	No effect.
1	Disable Busy Bank Interrupt.

Bit 15 – NAK_OUT NAKOUT Interrupt Disable

Value	Description
0	No effect.
1	Disable NAKOUT Interrupt.

Bit 14 – NAK_IN NAKIN Interrupt Disable

Value	Description
0	No effect.
1	Disable NAKIN Interrupt.

Bit 13 – STALL_SNT Stall Sent Interrupt Disable

Value	Description
0	No effect.
1	Disable Stall Sent Interrupt.

Bit 12 – RX_SETUP Received SETUP Interrupt Disable

Value	Description
0	No effect.
1	Disable RX_SETUP Interrupt.

Bit 11 – TXRDY TX Packet Ready Interrupt Disable

Value	Description
0	No effect.
1	Disable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Disable

Value	Description
0	No effect.
1	Disable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Disable

Value	Description
0	No effect.
1	Disable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Overflow Error Interrupt.

Bit 4 – NYET_DIS NYET Enable (Only for High Speed Bulk OUT endpoints)

Value	Description
0	No effect.
1	Let the hardware handle the handshake response for the High Speed Bulk OUT transfer.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	Disable the “Interrupts Disable DMA”.

Bit 1 – AUTO_VALID Packet Auto-Valid Disable

Value	Description
0	No effect.
1	Disable this bit to not automatically validate the current packet.

Bit 0 – EPT_DISABL Endpoint Disable

Value	Description
0	No effect.
1	Disable endpoint.

72.7.12 UDPHS Endpoint Control Disable Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLDISx
Offset: 0x0108 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

For additional information, see UDPHS_EPTCTLx.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
Access	W	W			W		W	W
Reset	–	–			–		–	–

Bit 31 – SHRT_PCKT Short Packet Interrupt Disable
For IN endpoints: Never automatically add a zero length packet at end of DMA transfer.
For OUT endpoints:

Value	Description
0	No effect.
1	Disable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Disable

Value	Description
0	No effect.
1	Disable Busy Bank Interrupt.

Bit 14 – ERR_FLUSH bank flush error Interrupt Disable

Value	Description
0	No effect.
1	Disable Bank Flush Error Interrupt.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Error CRC ISO/Error Number of Transaction Interrupt.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Disable

Value	Description
0	No effect.

Value	Description
1	Disable Error Flow ISO Interrupt.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Disable

Value	Description
0	No effect.
1	Disable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Disable

Value	Description
0	No effect.
1	Disable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Disable

Value	Description
0	No effect.
1	Disable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Overflow Error Interrupt.

Bit 7 – MDATA_RX MDATA Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Disable MDATA Interrupt.

Bit 6 – DATAX_RX DATAx Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Disable DATAx Interrupt.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	Disable the “Interrupts Disable DMA”.

Bit 1 – AUTO_VALID Packet Auto-Valid Disable

Value	Description
0	No effect.
1	Disable this bit to not automatically validate the current packet.

Bit 0 – EPT_DISABL Endpoint Disable

Value	Description
0	No effect.
1	Disable endpoint.

72.7.13 UDPHS Endpoint Control Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLx
Offset: 0x010C + x*0x20 [x=0..15]
Reset: 0x00000000
Property: Read-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

The reset value for UDPHS_EPTCTL0 is 0x00000001.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access				R	R		R	R
Reset				0	0		0	0

Bit 31 – SHRT_PCKT Short Packet Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: sends an Interrupt when a Short Packet has been received.
For IN endpoints: a Short Packet transmission is ensured upon end of the DMA Transfer, thus signaling a BULK or INTERRUPT end of transfer, but only if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTO_VALID bits are also set.

Value	Description
0	Short Packet Interrupt is masked.
1	Short Packet Interrupt is enabled.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: an interrupt is sent when all banks are busy.
For IN endpoints: an interrupt is sent when all banks are free.

Value	Description
0	BUSY_BANK Interrupt is masked.
1	BUSY_BANK Interrupt is enabled.

Bit 15 – NAK_OUT NAKOUT Interrupt Enabled (cleared upon USB reset)

Value	Description
0	NAKOUT Interrupt is masked.
1	NAKOUT Interrupt is enabled.

Bit 14 – NAK_IN NAKIN Interrupt Enabled (cleared upon USB reset)

Value	Description
0	NAKIN Interrupt is masked.
1	NAKIN Interrupt is enabled.

Bit 13 – STALL_SNT Stall Sent Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Stall Sent Interrupt is masked.
1	Stall Sent Interrupt is enabled.

Bit 12 – RX_SETUP Received SETUP Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received SETUP is masked.
1	Received SETUP is enabled.

Bit 11 – TXRDY TX Packet Ready Interrupt Enabled (cleared upon USB reset)

⚠ CAUTION Interrupt source is active as long as the corresponding UDPHS_EPTSTAx register TXRDY flag remains low. If there are no more banks available for transmitting after the software has set UDPHS_EPTSTAx/TXRDY for the last transmit packet, then the interrupt source remains inactive until the first bank becomes free again to transmit at UDPHS_EPTSTAx/TXRDY hardware clear.

Value	Description
0	TX Packet Ready Interrupt is masked.
1	TX Packet Ready Interrupt is enabled.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Transmitted IN Data Complete Interrupt is masked.
1	Transmitted IN Data Complete Interrupt is enabled.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received OUT Data Interrupt is masked.
1	Received OUT Data Interrupt is enabled.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Overflow Error Interrupt is masked.
1	Overflow Error Interrupt is enabled.

Bit 4 – NYET_DIS NYET Disable (Only for High Speed Bulk OUT Endpoints) (cleared upon USB reset)
Note: According to the Universal Serial Bus Specification, Rev 2.0 (8.5.1.1 NAK Responses to OUT/DATA During PING Protocol), a NAK response to an HS Bulk OUT transfer is expected to be an unusual occurrence.

Value	Description
0	Lets the hardware handle the handshake response for the High Speed Bulk OUT transfer.
1	Forces an ACK response to the next High Speed Bulk OUT transfer instead of a NYET response.

Bit 3 – INTDIS_DMA Interrupt Disables DMA (cleared upon USB reset)
If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled regardless of the UDPHS_IEN register EPT_x bit for this endpoint. Then, the firmware will have to clear or disable the interrupt source or clear this bit if transfer completion is needed.
If the exception raised is associated with the new system bank packet, then the previous DMA packet transfer is normally completed, but the new DMA packet transfer is not started (not requested).
If the exception raised is not associated to a new system bank packet (NAK_IN, NAK_OUT, etc.), then the request cancellation may happen at any time and may immediately stop the current DMA transfer.
This may be used, for example, to identify or prevent an erroneous packet to be transferred into a buffer or to complete a DMA buffer by software after reception of a short packet.

Bit 1 – AUTO_VALID Packet Auto-Valid Enabled (Not for CONTROL Endpoints) (cleared upon USB reset)
Set this bit to automatically validate the current packet and switch to the next bank for both IN and OUT endpoints.
For IN Transfer:
If this bit is set, the UDPHS_EPTSTAx register TXRDY bit is set automatically when the current bank is full and at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set.
The user may still set the UDPHS_EPTSTAx register TXRDY bit if the current bank is not full, unless the user needs to send a Zero Length Packet by software.
For OUT Transfer:
If this bit is set, the UDPHS_EPTSTAx register RXRDY_TXKL bit is automatically reset for the current bank when the last packet byte has been read from the bank FIFO or at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set. For example, to truncate a padded data packet when the actual data transfer size is reached.
The user may still clear the UDPHS_EPTSTAx register RXRDY_TXKL bit, for example, after completing a DMA buffer by software if UDPHS_DMACONTROLx register END_B_EN bit was disabled or in order to cancel the read of the remaining data bank(s).

Bit 0 – EPT_ENABL Endpoint Enable (cleared upon USB reset)

Value	Description
0	The endpoint is disabled according to the device configuration. Endpoint 0 should always be enabled after a hardware or UDPHS bus reset and participate in the device configuration.
1	The endpoint is enabled according to the device configuration.

72.7.14 UDPHS Endpoint Control Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLx
Offset: 0x010C + x*0x20 [x=0..15]
Reset: 0x00000000
Property: Read-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

The reset value for UDPHS_EPTCTL0 is 0x00000001.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access	R	R			R		R	R
Reset	0	0			0		0	0

Bit 31 – SHRT_PCKT Short Packet Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: Send an Interrupt when a Short Packet has been received.
For IN endpoints: A Short Packet transmission is ensured upon end of the DMA Transfer, thus signaling an end of isochronous (micro-)frame data, but only if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTO_VALID bits are also set.

Value	Description
0	Short Packet Interrupt is masked.
1	Short Packet Interrupt is enabled.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: An interrupt is sent when all banks are busy.
For IN endpoints: An interrupt is sent when all banks are free.

Value	Description
0	BUSY_BANK Interrupt is masked.
1	BUSY_BANK Interrupt is enabled.

Bit 14 – ERR_FLUSH Bank Flush Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Bank Flush Error Interrupt is masked.
1	Bank Flush Error Interrupt is enabled.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	ISO CRC error/number of Transaction Error Interrupt is masked.
1	ISO CRC error/number of Transaction Error Interrupt is enabled.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Error Flow Interrupt is masked.
1	Error Flow Interrupt is enabled.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Enabled (cleared upon USB reset)

⚠ CAUTION Interrupt source is active as long as the corresponding UDPHS_EPTSTAx register TXRDY_TRER flag remains low. If there are no more banks available for transmitting after the software has set UDPHS_EPTSTAx/TXRDY_TRER for the last transmit packet, then the interrupt source remains inactive until the first bank becomes free again to transmit at UDPHS_EPTSTAx/TXRDY_TRER hardware clear.

Value	Description
0	TX Packet Ready/Transaction Error Interrupt is masked.
1	TX Packet Ready/Transaction Error Interrupt is enabled.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Transmitted IN Data Complete Interrupt is masked.
1	Transmitted IN Data Complete Interrupt is enabled.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received OUT Data Interrupt is masked.
1	Received OUT Data Interrupt is enabled.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Overflow Error Interrupt is masked.
1	Overflow Error Interrupt is enabled.

Bit 7 – MDATA_RX MDATA Interrupt Enabled (Only for High Bandwidth Isochronous OUT endpoints) (cleared upon USB reset)

Value	Description
0	No effect.
1	Send an interrupt when an MDATA packet has been received and so at least one packet of the microframe data payload has been received.

Bit 6 – DATAX_RX DATAx Interrupt Enabled (Only for High Bandwidth Isochronous OUT endpoints) (cleared upon USB reset)

Value	Description
0	No effect.
1	Send an interrupt when a DATA2, DATA1 or DATA0 packet has been received meaning the whole microframe data payload has been received.

Bit 3 – INTDIS_DMA Interrupt Disables DMA (cleared upon USB reset)
If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled regardless of the UDPHS_IEN register EPT_x bit for this endpoint. Then, the firmware will have to clear or disable the interrupt source or clear this bit if transfer completion is needed.
If the exception raised is associated with the new system bank packet, then the previous DMA packet transfer is normally completed, but the new DMA packet transfer is not started (not requested).
If the exception raised is not associated to a new system bank packet (ex: ERR_FL_ISO), then the request cancellation may happen at any time and may immediately stop the current DMA transfer.
This may be used, for example, to identify or prevent an erroneous packet to be transferred into a buffer or to complete a DMA buffer by software after reception of a short packet, or to perform buffer truncation on ERR_FL_ISO interrupt for adaptive rate.

Bit 1 – AUTO_VALID Packet Auto-Valid Enabled (cleared upon USB reset)
Set this bit to automatically validate the current packet and switch to the next bank for both IN and OUT endpoints.
For IN Transfer:
If this bit is set, the UDPHS_EPTSTAx register TXRDY_TRER bit is set automatically when the current bank is full and at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set.
The user may still set the UDPHS_EPTSTAx register TXRDY_TRER bit if the current bank is not full, unless the user needs to send a Zero Length Packet by software.
For OUT Transfer:
If this bit is set, the UDPHS_EPTSTAx register RXRDY_TXKL bit is automatically reset for the current bank when the last packet byte has been read from the bank FIFO or at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set. For example, to truncate a padded data packet when the actual data transfer size is reached.
The user may still clear the UDPHS_EPTSTAx register RXRDY_TXKL bit, for example, after completing a DMA buffer by software if UDPHS_DMACONTROLx register END_B_EN bit was disabled or in order to cancel the read of the remaining data bank(s).

Bit 0 – EPT_ENABL Endpoint Enable (cleared upon USB reset)

Value	Description
0	The endpoint is disabled according to the device configuration. Endpoint 0 should always be enabled after a hardware or UDPHS bus reset and participate in the device configuration.
1	The endpoint is enabled according to the device configuration.

72.7.15 UDPHS Endpoint Set Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTSETSTAx
Offset: 0x0114 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

For additional information, see UDPHS_EPTSTAx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					TXRDY		RXRDY_TXKL	
Access					W		W	
Reset					–		–	

Bit	7	6	5	4	3	2	1	0
			FRCESTALL					
Access			W					
Reset			–					

Bit 11 – TXRDY TX Packet Ready Set

Value	Description
0	No effect.
1	Set this bit after a packet has been written into the endpoint FIFO for IN data transfers – This flag is used to generate a Data IN transaction (device to host). – Device firmware checks that it can write a data payload in the FIFO, checking that TXRDY is cleared. – Transfer to the FIFO is done by writing in the “Buffer Address” register. – Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TXRDY to one. – UDPHS bus transactions can start. – TXCOMP is set once the data payload has been received by the host. – Data should be written into the endpoint FIFO only after this bit has been cleared. – Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

Bit 9 – RXRDY_TXKL KILL Bank Set (for IN Endpoint)

Value	Description
0	No effect.
1	Kill the last written bank.

Bit 5 – FRCESTALL Stall Handshake Request Set
Refer to chapters 8.4.5 (Handshake Packets) and 9.4.5 (Get Status) of the Universal Serial Bus Specification, Rev 2.0 for more information on the STALL handshake.

Value	Description
0	No effect.
1	Set this bit to request a STALL answer to the host for the next handshake

72.7.16 UDPHS Endpoint Set Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSETSTAx
Offset: 0x0114 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

For additional information, see UDPHS_EPTSTAx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					TXRDY_TRER		RXRDY_TXKL	
Access					W		W	
Reset					–		–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 11 – TXRDY_TRER TX Packet Ready Set

Value	Description
0	No effect.
1	Set this bit after a packet has been written into the endpoint FIFO for IN data transfers – This flag is used to generate a Data IN transaction (device to host). – Device firmware checks that it can write a data payload in the FIFO, checking that TXRDY_TRER is cleared. – Transfer to the FIFO is done by writing in the “Buffer Address” register. – Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TXRDY_TRER to one. – UDPHS bus transactions can start. – TXCOMP is set once the data payload has been sent. – Data should be written into the endpoint FIFO only after this bit has been cleared. – Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

Bit 9 – RXRDY_TXKL KILL Bank Set (for IN Endpoint)

Value	Description
0	No effect.
1	Kill the last written bank.

72.7.17 UDPHS Endpoint Clear Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCLRSTAx
Offset: 0x0118 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

For additional information, see UDPHS_EPTSTAx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
Access	W	W	W	W		W	W	
Reset	–	–	–	–		–	–	

Bit	7	6	5	4	3	2	1	0
		TOGGLESQ	FRCESTALL					
Access		W	W					
Reset		–	–					

Bit 15 – NAK_OUT NAKOUT Clear

Value	Description
0	No effect.
1	Clear the NAK_OUT flag of UDPHS_EPTSTAx.

Bit 14 – NAK_IN NAKIN Clear

Value	Description
0	No effect.
1	Clear the NAK_IN flags of UDPHS_EPTSTAx.

Bit 13 – STALL_SNT Stall Sent Clear

Value	Description
0	No effect.
1	Clear the STALL_SNT flags of UDPHS_EPTSTAx.

Bit 12 – RX_SETUP Received SETUP Clear

Value	Description
0	No effect.
1	Clear the RX_SETUP flags of UDPHS_EPTSTAx.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Clear

Value	Description
0	No effect.
1	Clear the TX_COMPLT flag of UDPHS_EPTSTAx.

Bit 9 – RXRDY_TXKL Received OUT Data Clear

Value	Description
0	No effect.
1	Clear the RXRDY_TXKL flag of UDPHS_EPTSTAx.

Bit 6 – TOGGLESQ Data Toggle Clear
For OUT endpoints, the next received packet should be a DATA0.
For IN endpoints, the next packet will be sent with a DATA0 PID.

Value	Description
0	No effect.
1	Clear the PID data of the current bank

Bit 5 – FRCESTALL Stall Handshake Request Clear

Value	Description
0	No effect.
1	Clear the STALL request. The next packets from host will not be STALLed.

72.7.18 UDPHS Endpoint Clear Status Register (Isochronous Endpoint)

Name: UDPHS_EPTCLRSTAx
Offset: 0x0118 + x*0x20 [x=0..15]
Reset: –
Property: Write-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

For additional information, see UDPHS_EPTSTAx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
Access		W	W	W		W	W	
Reset		–	–	–		–	–	

Bit	7	6	5	4	3	2	1	0
		TOGGLESQ						
Access		W						
Reset		–						

Bit 14 – ERR_FLUSH Bank Flush Error Clear

Value	Description
0	No effect.
1	Clear the ERR_FLUSH flags of UDPHS_EPTSTAx.

Bit 13 – ERR_CRC_NTR Number of Transaction Error Clear

Value	Description
0	No effect.
1	Clear the ERR_CRC_NTR flags of UDPHS_EPTSTAx.

Bit 12 – ERR_FL_ISO Error Flow Clear

Value	Description
0	No effect.
1	Clear the ERR_FL_ISO flags of UDPHS_EPTSTAx.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Clear

Value	Description
0	No effect.
1	Clear the TX_COMPLT flag of UDPHS_EPTSTAx.

Bit 9 – RXRDY_TXKL Received OUT Data Clear

Value	Description
0	No effect.
1	Clear the RXRDY_TXKL flag of UDPHS_EPTSTAx.

Bit 6 – TOGGLESQ Data Toggle Clear
For OUT endpoints, the next received packet should be a DATA0.
For IN endpoints, the next packet will be sent with a DATA0 PID.

Value	Description
0	No effect.
1	Clear the PID data of the current bank

72.7.19 UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTSTAx
Offset: 0x011C + x*0x20 [x=0..15]
Reset: 0x00000040
Property: Read-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x0, 0x2 or 0x3.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT	BYTE_COUNT[10:4]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TOGGLESQ_STA[1:0]		FRCESTALL					
Access	R	R	R					
Reset	0	1	0					

Bit 31 – SHRT_PCKT Short Packet (cleared upon USB reset)
An OUT Short Packet is detected when the receive byte count is less than the configured UDPHS_EPTCFGx register EPT_Size.
This bit is updated at the same time as the BYTE_COUNT field.
It is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 30:20 – BYTE_COUNT[10:0] UDPHS Byte Count (cleared upon USB reset)
Byte count of a received data packet.
This field is incremented after each write into the endpoint (to prepare an IN transfer). It is decremented after each reading into the endpoint (OUT transfer).
This field is also updated at RXRDY_TXKL flag clear with the next bank, and at TXRDY flag set with the next bank.
This field is reset by UDPHS_EPTRST.EPT_x.

Bits 19:18 – BUSY_BANK_STA[1:0] Busy Bank Number (cleared upon USB reset)
These bits are set by hardware to indicate the number of busy banks.
IN endpoint: Indicates the number of busy banks filled by the user, ready for IN transfer.
OUT endpoint: Indicates the number of busy banks filled by OUT transaction from the Host.

Value	Name	Description
0	0BUSYBANK	All banks are free
1	1BUSYBANK	1 busy bank
2	2BUSYBANKS	2 busy banks
3	3BUSYBANKS	3 busy banks

Bits 17:16 – CURBK_CTLDIR[1:0] Current Bank/Control Direction (cleared upon USB reset)
Control Direction (for Control endpoint only):
0: A Control Write is requested by the Host.
1: A Control Read is requested by the Host.

Notes:
1. Corresponds to the the 7th bit of the bmRequestType (Byte 0 of the Setup Data).
2. Updated after receiving new setup data.

Current Bank (not relevant for Control endpoint):
- Set by hardware to indicate the number of the current bank.
- Reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).
- The current bank is updated each time the user:
  – Sets the TX Packet Ready bit to prepare the next IN transfer and to switch to the next bank.
  – Clears the received OUT data bit to access the next bank.

Value	Name	Description
0	BANK0	Bank 0 (or single bank)
1	BANK1	Bank 1
2	BANK2	Bank 2

Bit 15 – NAK_OUT NAK OUT (cleared upon USB reset)
This bit is set by hardware when a NAK handshake has been sent in response to an OUT or PING request from the Host.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by EPT_CTL_DISx (disable endpoint).

Bit 14 – NAK_IN NAK IN (cleared upon USB reset)
This bit is set by hardware when a NAK handshake has been sent in response to an IN request from the Host.
This bit is cleared by software.

Bit 13 – STALL_SNT Stall Sent (cleared upon USB reset)
For Control, Bulk and Interrupt endpoints.
This bit is set by hardware after a STALL handshake has been sent as requested by the UDPHS_EPTSTAx register FRCESTALL bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 12 – RX_SETUP Received SETUP (cleared upon USB reset)
For Control endpoint only.
This bit is set by hardware when a valid SETUP packet has been received from the host.
It is cleared by the device firmware after reading the SETUP data from the endpoint FIFO.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 11 – TXRDY TX Packet Ready (cleared upon USB reset)
This bit is cleared by hardware after the host has acknowledged the packet.
For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.
Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 10 – TX_COMPLT Transmitted IN Data Complete (cleared upon USB reset)
This bit is set by hardware after an IN packet has been accepted (ACK'ed) by the host.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 9 – RXRDY_TXKL Received OUT Data/KILL Bank (cleared upon USB reset)
Received OUT Data (for OUT endpoint or Control endpoint):
- This bit is set by hardware after a new packet has been stored in the endpoint FIFO.
- This bit is cleared by the device firmware after reading the OUT data from the endpoint.
- For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.

- Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.
- This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

KILL Bank (for IN endpoint):

- The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.
- The bank is not cleared but sent on the IN transfer, TX_COMPLT
- The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.
  Note: "Kill a packet" may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

Bit 8 – ERR_OVFLW Overflow Error (cleared upon USB reset)
This bit is set by hardware when a new too-long packet is received.
Example: If the user programs an endpoint 64 bytes wide and the host sends 128 bytes in an OUT transfer, then the Overflow Error bit is set.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 7:6 – TOGGLESQ_STA[1:0] Toggle Sequencing (cleared upon USB reset)
In OUT transfer, the Toggle information is meaningful only when the current bank is busy (Received OUT Data = 1).
This field is updated for OUT transfer:

- A new data has been written into the current bank.
- The user has just cleared the Received OUT Data bit to switch to the next bank.

This field is reset to DATA1 by the UDPHS_EPTCLRSTAx register TOGGLESQ bit, and by UDPHS_EPTCTLDISx (disable endpoint).
Toggle Sequencing:

- IN endpoint: Indicates the PID Data Toggle that will be used for the next packet sent. This is not relative to the current bank.
- CONTROL and OUT endpoints: Set by hardware to indicate the PID data of the current bank.

Value	Name	Description
0	DATA0	DATA0
1	DATA1	DATA1
2	DATA2	Reserved for High Bandwidth Isochronous Endpoint
3	MDATA	Reserved for High Bandwidth Isochronous Endpoint

Bit 5 – FRCESTALL Stall Handshake Request (cleared upon USB reset)
This bit is reset by hardware upon received SETUP.

Value	Description
0	No effect.
1	If set a STALL answer will be done to the host for the next handshake.

72.7.20 UDPHS Endpoint Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSTAx
Offset: 0x011C + x*0x20 [x=0..15]
Reset: 0x00000040
Property: Read-only

This register view is relevant only if UDPHS_EPTCFGx.EPT_TYPE = 0x1.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT	BYTE_COUNT[10:4]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TOGGLESQ_STA[1:0]							
Access	R	R						
Reset	0	1						

Bit 31 – SHRT_PCKT Short Packet (cleared upon USB reset)
An OUT Short Packet is detected when the receive byte count is less than the configured UDPHS_EPTCFGx register EPT_Size.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 30:20 – BYTE_COUNT[10:0] UDPHS Byte Count (cleared upon USB reset)
Byte count of a received data packet.
This field is incremented after each write into the endpoint (to prepare an IN transfer).
This field is decremented after each reading into the endpoint (OUT transfer).
This field is also updated at RXRDY_TXKL flag clear with the next bank.
This field is also updated at TXRDY_TRER flag set with the next bank.
This field is reset by EPT_x of UDPHS_EPTRST register.

Bits 19:18 – BUSY_BANK_STA[1:0] Busy Bank Number (cleared upon USB reset)
These bits are set by hardware to indicate the number of busy banks.
IN endpoint: It indicates the number of busy banks filled by the user, ready for IN transfer.
OUT endpoint: It indicates the number of busy banks filled by OUT transaction from the Host.

Value	Name	Description
0	0BUSYBANK	All banks are free
1	1BUSYBANK	1 busy bank
2	2BUSYBANKS	2 busy banks
3	3BUSYBANKS	3 busy banks

Bits 17:16 – CURBK[1:0] Current Bank (cleared upon USB reset)
These bits are set by hardware to indicate the number of the current bank.

The current bank is updated each time the user:

- Sets the TX Packet Ready bit to prepare the next IN transfer and to switch to the next bank.
- Clears the received OUT data bit to access the next bank.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Value	Name	Description
0	BANK0	Bank 0 (or single bank)
1	BANK1	Bank 1
2	BANK2	Bank 2

Bit 14 – ERR_FLUSH Bank Flush Error (cleared upon USB reset)
For High Bandwidth Isochronous IN endpoints.
This bit is set when flushing unsent banks at the end of a microframe.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by EPT_CTL_DISx (disable endpoint).

Bit 13 – ERR_CRC_NTR CRC ISO Error/Number of Transaction Error (cleared upon USB reset)
CRC ISO Error (for Isochronous OUT endpoints) (Read-only):
This bit is set by hardware if the last received data is corrupted (CRC error on data).
This bit is updated by hardware when new data is received (Received OUT Data bit).
Number of Transaction Error (for High Bandwidth Isochronous IN endpoints):
This bit is set at the end of a microframe in which at least one data bank has been transmitted, if less than the number of transactions per micro-frame banks (UDPHS_EPTCFGx register NB_TRANS) have been validated for transmission inside this microframe.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 12 – ERR_FL_ISO Error Flow (cleared upon USB reset)
This bit is set by hardware when a transaction error occurs.

- Isochronous IN transaction is missed, the micro has no time to fill the endpoint (underflow).
- Isochronous OUT data is dropped because the bank is busy (overflow).

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error (cleared upon USB reset)
TX Packet Ready
This bit is cleared by hardware, as soon as the packet has been sent.
For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.
Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY_TRER bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).
Transaction Error (for high bandwidth isochronous OUT endpoints) (Read-Only):
This bit is set by hardware when a transaction error occurs inside one microframe.
If one toggle sequencing problem occurs among the n-transactions (n = 1, 2 or 3) inside a microframe, then this bit is still set as long as the current bank contains one “bad” n-transaction (see CURBK field description). As soon as the current bank is relative to a new “good” n-transactions, then this bit is reset.
Notes:

1. A transaction error occurs when the toggle sequencing does not comply with the Universal Serial Bus Specification, Rev 2.0 (5.9.2 High Bandwidth Isochronous endpoints) (Bad PID, missing data, etc.)
2. When a transaction error occurs, the user may empty all the “bad” transactions by clearing the Received OUT Data flag (RXRDY_TXKL).

If this bit is reset, then the user should consider that a new n-transaction is coming.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 10 – TX_COMPLT Transmitted IN Data Complete (cleared upon USB reset)
This bit is set by hardware after an IN packet has been sent.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 9 – RXRDY_TXKL Received OUT Data/KILL Bank (cleared upon USB reset)
Received OUT Data (for OUT endpoint or Control endpoint):

- This bit is set by hardware after a new packet has been stored in the endpoint FIFO.
- This bit is cleared by the device firmware after reading the OUT data from the endpoint.
- For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.
- Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.
- This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

KILL Bank (for IN endpoint):

- The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.
- The bank is not cleared but sent on the IN transfer, TX_COMPLT
- The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.
  Note: “Kill a packet” may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

Bit 8 – ERR_OVFLW Overflow Error (cleared upon USB reset)
This bit is set by hardware when a new too-long packet is received.
Example: If the user programs an endpoint 64 bytes wide and the host sends 128 bytes in an OUT transfer, then the Overflow Error bit is set.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 7:6 – TOGGLESQ_STA[1:0] Toggle Sequencing (cleared upon USB reset)
In OUT transfer, the Toggle information is meaningful only when the current bank is busy (Received OUT Data = 1).
This field is updated for OUT transfer:

- A new data has been written into the current bank.
- The user has just cleared the Received OUT Data bit to switch to the next bank.

For High Bandwidth Isochronous Out endpoint, it is recommended to check the UDPHS_EPTSTAx/TXRDY_TRER bit to know if the toggle sequencing is correct or not.
This field is reset to DATA1 by the UDPHS_EPTCLRSTAx register TOGGLESQ bit, and by UDPHS_EPTCTLDISx (disable endpoint).
Toggle Sequencing:

- IN endpoint: Indicates the PID Data Toggle that will be used for the next packet sent. This is not relative to the current bank.
- OUT endpoint: Set by hardware to indicate the PID data of the current bank:

Value	Name	Description
0	DATA0	DATA0
1	DATA1	DATA1
2	DATA2	Data2 (only for High Bandwidth Isochronous Endpoint)
3	MDATA	MData (only for High Bandwidth Isochronous Endpoint)

72.7.21 UDPHS DMA Channel Transfer Descriptor

The DMA channel transfer descriptor is loaded from the memory. Be careful with the alignment of this buffer. The structure of the DMA channel transfer descriptor is defined by three parameters as described below:

- Offset 0:
  – The address must be aligned: 0xXXXX0
  – Next Descriptor Address Register: UDPHS_DMANXTDSCx
- Offset 4:
  – The address must be aligned: 0xXXXX4
  – DMA Channelx Address Register: UDPHS_DMAADDRESSx
- Offset 8:
  – The address must be aligned: 0xXXXX8
  – DMA Channelx Control Register: UDPHS_DMACONTROLx

To use the DMA channel transfer descriptor, fill the structures with the correct value (as described in the following pages). Then write directly in UDPHS_DMANXTDSCx the address of the descriptor to be used first. Then write '1' in the LDNXT_DSC bit of UDPHS_DMACONTROLx (load next channel transfer descriptor). The descriptor is automatically loaded upon Endpointx request for packet transfer.

72.7.22 UDPHS DMA Next Descriptor Address Register

Name: UDPHS_DMANXTDSCx
Offset: 0x0310 + (x-1)*0x10 [x=1..7]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	NXT_DSC_ADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NXT_DSC_ADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NXT_DSC_ADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NXT_DSC_ADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NXT_DSC_ADD[31:0] Next Descriptor Address
This field points to the next channel descriptor to be processed. This channel descriptor must be aligned, so bits 0 to 3 of the address must be equal to zero.

72.7.23 UDPHS DMA Channel Address Register

Name: UDPHS_DMAADDRESSx
Offset: 0x0314 + (x-1)*0x10 [x=1..7]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	BUFF_ADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_ADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BUFF_ADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUFF_ADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BUFF_ADD[31:0] Buffer Address
This field determines the system bus starting address of a DMA channel transfer.
Channel start and end addresses may be aligned on any byte boundary.
The firmware may write this field only when the UDPHS_DMASTATUS.CHANN_ENB bit is clear.
This field is updated at the end of the address phase of the current access to the system bus. It is incrementing of the access byte width. The access width is 4 bytes (or less) at packet start or end, if the start or end address is not aligned on a word boundary.
The packet start address is either the channel start address or the next channel address to be accessed in the channel buffer.
The packet end address is either the channel end address or the latest channel address accessed in the channel buffer.
The channel start address is written by software or loaded from the descriptor, whereas the channel end address is either determined by the end of buffer or the UDPHS device, USB end of transfer if the UDPHS_DMACONTROL.END_TR_EN bit is set.

72.7.24 UDPHS DMA Channel Control Register

Name: UDPHS_DMACONTROLx
Offset: 0x0318 + (x-1)*0x10 [x=1..7]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bits [31:2] are only writable when issuing a channel Control Command other than “Stop Now”.

For reliability it is highly recommended to wait for both UDPHS_DMASTATUS.CHAN_ACT and CHAN_ENB flags at 0, thus ensuring the channel has been stopped before issuing a command other than “Stop Now”.

Bit	31	30	29	28	27	26	25	24
	BUFF_LENGTH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	BUFF_LENGTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Access								
Reset								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – BUFF_LENGTH[15:0] Buffer Byte Length (Write-only)
This field determines the number of bytes to be transferred until end of buffer. The maximum channel transfer size (64 Kbytes) is reached when this field is 0 (default value). If the transfer size is unknown, this field should be set to 0, but the transfer end may occur earlier under UDPHS device control.
When this field is written, the UDPHS_DMASTATUS.BUFF_COUNT field is updated with the write value.

Bit 7 – BURST_LCK Burst Lock Enable

Value	Description
0	The DMA never locks bus access.
1	USB packets system data bursts are locked for maximum optimization of the bus bandwidth usage and maximization of fly-by system bus burst duration.

Bit 6 – DESC_LD_IT Descriptor Loaded Interrupt Enable

Value	Description
0	UDPHS_DMASTATUS.DESC_LDST rising will not trigger any interrupt.
1	An interrupt is generated when a descriptor has been loaded from the bus.

Bit 5 – END_BUFFIT End of Buffer Interrupt Enable

Value	Description
0	UDPHS_DMASTATUS.END_BF_ST rising will not trigger any interrupt.
1	An interrupt is generated when the UDPHS_DMASTATUS.BUFF_COUNT reaches zero.

Bit 4 – END_TR_IT End of Transfer Interrupt Enable

Value	Description
0	UDPHS device initiated buffer transfer completion will not trigger any interrupt at UDPHS_STATUS.END_TR_ST rising.
1	An interrupt is sent after the buffer transfer is complete, if the UDPHS device has ended the buffer transfer. Use when the receive size is unknown.

Bit 3 – END_B_EN End of Buffer Enable (Control)

Value	Description
0	DMA Buffer End has no impact on USB packet transfer.
1	Endpoint can validate the packet (according to the values programmed in the UDPHS_EPTCTL.AUTO_VALID and SHRT_PCKT fields) at DMA Buffer End, i.e., when the UDPHS_DMASTATUS.BUFF_COUNT reaches 0. This is mainly for short packet IN validation initiated by the DMA reaching end of buffer, but could be used for OUT packet truncation (discarding of unwanted packet data) at the end of DMA buffer.

Bit 2 – END_TR_EN End of Transfer Enable (Control)
Used for OUT transfers only.

Value	Description
0	USB end of transfer is ignored.
1	UDPHS device can put an end to the current buffer transfer. When set, a BULK or INTERRUPT short packet or the last packet of an ISOCHRONOUS (micro) frame (DATAX) will close the current buffer and the UDPHS_DMASTATUS.END_TR_ST flag will be raised. This is intended for UDPHS non-prenegotiated end of transfer (BULK or INTERRUPT) or ISOCHRONOUS microframe data buffer closure.

Bit 1 – LDNXT_DSC Load Next Channel Transfer Descriptor Enable (Command)
If the CHANN_ENB bit is cleared, the next descriptor is immediately loaded upon transfer request.
DMA Channel Control Command Summary:

LDNXT_DSC	CHANN_ENB	Description
0	0	Stop now
0	1	Run and stop at end of buffer
1	0	Load next descriptor now
1	1	Run and link at end of buffer

Value	Description
0	No channel register is loaded after the end of the channel transfer.
1	The channel controller loads the next descriptor after the end of the current transfer, i.e., when the UDPHS_DMASTATUS.CHANN_ENB bit is reset.

Bit 0 – CHANN_ENB (Channel Enable Command)

Value	Description
0	DMA channel is disabled at and no transfer will occur upon request. This bit is also cleared by hardware when the channel source bus is disabled at end of buffer. If the UDPHS_DMACONTROL.LDNXT_DSC bit has been cleared by descriptor loading, the firmware will have to set the corresponding CHANN_ENB bit to start the described transfer, if needed. If the LDNXT_DSC bit is cleared, the channel is frozen and the channel registers may then be read and/or written reliably as soon as both UDPHS_DMASTATUS.CHANN_ENB and CHANN_ACT flags read as 0. If a channel request is currently serviced when this bit is cleared, the DMA FIFO buffer is drained until it is empty, then the UDPHS_DMASTATUS.CHANN_ENB bit is cleared. If the LDNXT_DSC bit is set at or after this bit clearing, then the currently loaded descriptor is skipped (no data transfer occurs) and the next descriptor is immediately loaded.

Value	Description
1	The UDPHS_DMASTATUS.CHANN_ENB bit will be set, thus enabling DMA channel data transfer. Then, any pending request will start the transfer. This may be used to start or resume any requested transfer.

72.7.25 UDPHS DMA Channel Status Register

Name: UDPHS_DMASTATUSx
Offset: 0x031C + (x-1)*0x10 [x=1..7]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	BUFF_COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 31:16 – BUFF_COUNT[15:0] Buffer Byte Count
This field determines the current number of bytes still to be transferred for this buffer. It is decremented from the source system bus access byte width at the end of this bus address phase.
The access byte width is 4 by default, or less, at DMA start or end, if the start or end address is not aligned on a word boundary.
At the end of buffer, the DMA accesses the UDPHS device only for the number of bytes needed to complete it.
This field value is reliable (stable) only if the channel has been stopped or frozen (the UDPHS_EPTCTLx.NT_DIS_DMA bit is used to disable the channel request) and the channel is no longer active (CHANN_ACT flag is 0).
Note: For OUT endpoints, if the receive buffer byte length (BUFF_LENGTH) has been defaulted to zero because the USB transfer length is unknown, the actual buffer byte length received is 0x10000-BUFF_COUNT.

Bit 6 – DESC_LDST Descriptor Loaded Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when a descriptor has been loaded from the system bus.

Bit 5 – END_BF_ST End of Channel Buffer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when the BUFF_COUNT countdown reaches zero.

Bit 4 – END_TR_ST End of Channel Transfer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.

Value	Description
1	Set by hardware when the last packet transfer is complete, if the UDPHS device has ended the transfer.

Bit 1 – CHANN_ACT Channel Active Status
When a packet transfer is ended, this bit is automatically reset.
When a packet transfer cannot be completed due to an END_BF_ST, this flag stays set during the next channel descriptor load (if any) and potentially until UDPHS packet transfer completion, if allowed by the new descriptor.

Value	Description
0	The DMA channel is no longer trying to source the packet data.
1	The DMA channel is currently trying to source packet data, i.e., selected as the highest-priority requesting channel.

Bit 0 – CHANN_ENB Channel Enable Status
When any transfer is ended either due to an elapsed byte count or a UDPHS device initiated transfer end, this bit is automatically reset.
This bit is normally set or cleared by writing into the UDPHS_DMACONTROLx.CHANN_ENB bit either by software or descriptor loading.
If a channel request is currently serviced when the CHANN_ENB bit is cleared, the DMA FIFO buffer is drained until it is empty, then this status bit is cleared.

Value	Description
0	The DMA channel no longer transfers data, and may load the next descriptor if the UDPHS_DMACONTROLx.LDNXT_DSC bit is set.
1	The DMA channel is currently enabled and transfers data upon request.

73. USB Host High Speed Port (UHPHS)

73.1 Description

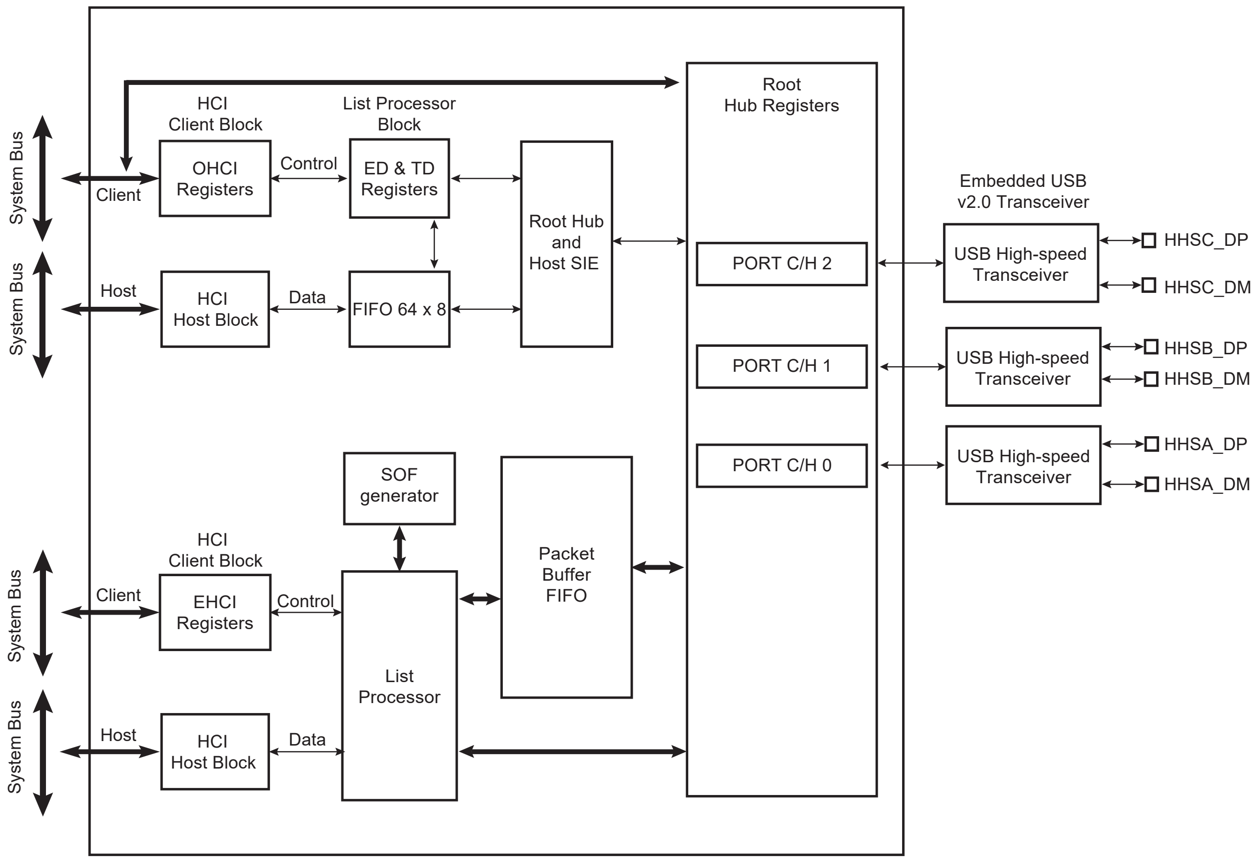
The USB Host High Speed Port (UHPHS) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as Enhanced HCI protocol (Enhanced Host Controller Interface).

73.2 Embedded Characteristics

- Compliant with Enhanced HCI Rev 1.0 Specification
  - Compliant with USB V2.0 High-speed Specification
  - Supports high-speed 480 Mbps
- Compliant with Open HCI Rev 1.0 Specification
  - Compliant with USB V2.0 Full-speed and Low-speed Specification
  - Supports both low-speed 1.5 Mbps and full-speed 12 Mbps USB devices
- Root Hub Integrated with Downstream USB HS Ports
- Embedded USB Transceivers
- Supports Power Management
- Three High-Speed (EHCI) and Full-Speed (OHCI) Hosts (A, B and C)
- Two High-Speed Devices Shared with Hosts A and B

73.3 Block Diagram

Figure 73-1. UHPHS Block Diagram



Access to the USB host operational registers is achieved through the system bus client interface. The Open HCI host controller and Enhanced HCI host controller initialize host DMA transfers through the system bus host interface as follows:

- Fetches endpoint descriptors and transfer descriptors
- Accesses endpoint data from system memory
- Accesses HC communication area
- Writes status and retires transfer descriptor

Memory access errors (abort, misalignment) lead to an “Unrecoverable Error” indicated by the corresponding flag in the host controller operational registers.

The USB root hub is integrated in the USB host. Several USB downstream ports are available. The number of downstream ports can be determined by the software driver reading the root hub’s operational registers. Device connection is automatically detected by the USB host port logic.

USB physical transceivers are integrated in the product and driven by the root hub’s ports.

73.4 Typical Connection

For information on a typical connection, refer to the section “USB 2.0 PHY”.

73.5 Product Dependencies

73.5.1 I/O Lines

For information on I/O lines, refer to the section "USB 2.0 PHY".

73.5.2 Power Management

The system embeds three transceivers.

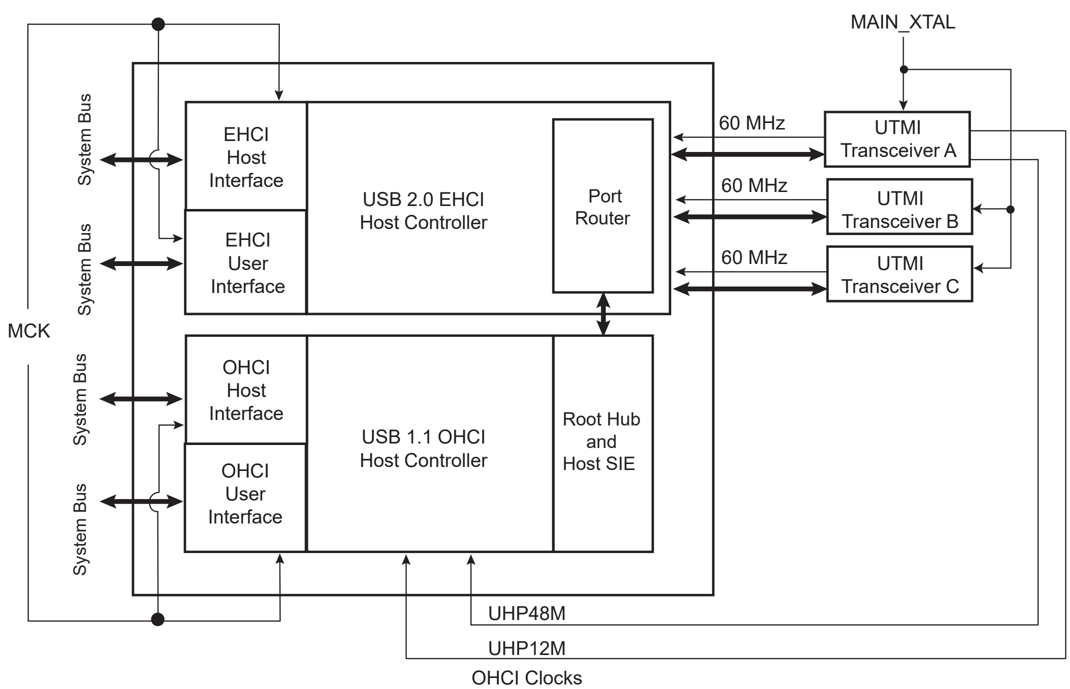
This clock is provided by the UTMI PLL embedded in each USB 2.0 PHY.

The High-speed transceiver returns a 60 MHz clock to the USB Host controller.

The USB Host controller requires 48 MHz and 12 MHz clocks for OHCI full-speed operations. These clocks must be generated by a PLL with a correct accuracy of +/-0.25% using the USBDIV field.

Thus the USB Host peripheral receives three clocks from the Power Management Controller (PMC): the Peripheral Clock (MCK domain), the UHP48M and the UHP12M (built-in UHP48M divided by four) used by the OHCI to interface with the bus USB signals (recovered 12 MHz domain) in Full-speed operations.

Figure 73-2. UHP Clock Trees



73.5.3 Interrupt Sources

The USB host interface has an interrupt line connected to the interrupt controller.

Handling USB host interrupts requires programming the interrupt controller before configuring the UHPHS.

73.6 Functional Description

73.6.1 UTMI Transceivers Sharing

For information on UTMI transceiver sharing, refer to the section "USB 2.0 PHY".

73.6.2 EHCI

The USB Host Port controller is fully compliant with the Enhanced HCI specification. The USB Host Port User Interface (registers description) can be found in the Enhanced HCI Rev 1.0 Specification available on www.usb.org

73.6.3 OHCI

The USB Host Port integrates a root hub and transceivers on downstream ports. It provides several Full-speed half-duplex serial communication ports at a baud rate of 12 Mbps. Up to 127 USB devices (printer, camera, mouse, keyboard, disk, etc.) and the USB hub can be connected to the USB host in the USB “tiered star” topology.

The USB Host Port controller is fully compliant with the Open HCI specification. The USB Host Port User Interface (registers description) can be found in the Open HCI Rev 1.0 Specification available on www.usb.org.

All standard class devices are automatically detected and available to the user’s application. As an example, integrating an HID (Human Interface Device) class driver provides a plug & play feature for all USB keyboards and mouses.

73.7 Register Summary

The Enhanced USB Host Controller contains two sets of software-accessible hardware registers: memory-mapped Host Controller Registers and optional PCI configuration registers. Note that the PCI configuration registers are only needed for PCI devices that implement the Host Controller.

- Memory-mapped USB Host Controller Registers—This block of registers is memory-mapped into non-cacheable memory. This memory space must begin on a DWord (32-bit) boundary. This register space is divided into two sections: a set of read-only capability registers and a set of read/write operational registers. The table below describes each register space.
  Note: Host controllers are not required to support exclusive-access mechanisms (such as PCI LOCK) for accesses to the memory-mapped register space. Therefore, if software attempts exclusive-access mechanisms to the host controller memory-mapped register space, the results are undefined.
- PCI Configuration Registers (for PCI devices)—In addition to the normal PCI header, power management, and device-specific registers, two registers are needed in the PCI configuration space to support USB. The normal PCI header and device-specific registers are beyond the scope of this document (the UHPHS_CLASSC register is shown in this document). Note that HCD does not interact with the PCI configuration space. This space is used only by the PCI enumerator to identify the USB Host Controller, and assign the appropriate system resources.
  The table below summarizes the enhanced interface register sets.

Offset	Register Set	Explanation
0 to N-1	Capability Registers	The capability registers specify the limits, restrictions, and capabilities of a host controller implementation. These values are used as parameters to the host controller driver.
N to N+M-1	Operational Registers	The operational registers are used by system software to control and monitor the operational state of the host controller.

Note: Software must not modify reserved bits in Read/Write registers.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	UHPHS_HCCAPBASE	31:24	HCIVERSION[15:8]							
		23:16	HCIVERSION[7:0]							
		15:8								
		7:0	CAPLENGTH[7:0]							
0x04	UHPHS_HCSPARAMS	31:24								
		23:16	N_DP[3:0]							P_INDICATOR
		15:8	N_CC[3:0]				N_PCC[3:0]			
		7:0				PPC	N_PORTS[3:0]			
0x08	UHPHS_HCCPARAMS	31:24								
		23:16								
		15:8	EECP[7:0]							
		7:0	IST[3:0]					ASPC	PFLF	AC
0x0C ... 0x0F	Reserved									
0x10	UHPHS_USBCMD	31:24								
		23:16	ITC[7:0]							
		15:8					ASPME		ASPMC[1:0]	
		7:0	LHCR	IAAD	ASE	PSE	FLS[1:0]		HCRESET	RS
0x14	UHPHS_USBSTS	31:24								
		23:16								
		15:8	ASS	PSS	RCM	HCHLT				
		7:0			IAA	HSE	FLR	PCD	USBERRINT	USBINT
0x18	UHPHS_USBINTR	31:24								
		23:16								
		15:8								
		7:0			IAAE	HSEE	FLRE	PCIE	USBEIE	USBIE
0x1C	UHPHS_FRINDEX	31:24								
		23:16								
		15:8			FI[13:8]					
		7:0	FI[7:0]							
0x20 ... 0x23	Reserved									
0x24	UHPHS_PERIODICLISTBASE	31:24	BA[19:12]							
		23:16	BA[11:4]							
		15:8	BA[3:0]							
		7:0								
0x28	UHPHS_ASYNCLISTADDR	31:24	LPL[26:19]							
		23:16	LPL[18:11]							
		15:8	LPL[10:3]							
		7:0	LPL[2:0]							
0x2C ... 0x4F	Reserved									

...........continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x50	UHPHS_CONFIGFLAG	31:24								
		23:16								
		15:8								
		7:0								CF
0x54	UHPHS_PORTSC0	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x58	UHPHS_PORTSC1	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x5C	UHPHS_PORTSC2	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x60 ... 0xA7	Reserved									
0xA8	UHPHS_INSNREG06	31:24	AHB_ERR							
		23:16								
		15:8					HBURST[2:0]			Nb_Burst[4]
		7:0	Nb_Burst[3:0]				Nb_Success_Burst[3:0]			
0xAC	UHPHS_INSNREG07	31:24	AHB_ADDR[31:24]							
		23:16	AHB_ADDR[23:16]							
		15:8	AHB_ADDR[15:8]							
		7:0	AHB_ADDR[7:0]							

73.7.1 UHPHS Host Controller Capability Register

Name: UHPHS_HCCAPBASE
Offset: 0x00
Reset: 0x01000010
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	HCIVERSION[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	HCIVERSION[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CAPLENGTH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 31:16 – HCIVERSION[15:0] Host Controller Interface Version Number
This is a two-byte field containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this field represents the major revision and the least significant byte the minor revision.

Bits 7:0 – CAPLENGTH[7:0] Capability Registers Length
This field is used as an offset to add to the register base to find the beginning of the Operational Register Space.

73.7.2 UHPHS Host Controller Structural Parameters Register

Name: UHPHS_HCSPARAMS
Offset: 0x04
Reset: 0x00001303
Property: Read-only

These fields define structural parameters: number of downstream ports, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	N_DP[3:0]							P_INDICATOR
Access	R	R	R	R				R
Reset	0	0	0	0				0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	N_CC[3:0]				N_PCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	1	1
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				PPC	N_PORTS[3:0]			
Access				R	R	R	R	R
Reset				0	0	0	1	1

Bits 23:20 – N_DP[3:0] Debug Port Number
Optional. This register identifies which of the host controller ports is the debug port. The value is the port number (1-based) of the debug port. A non-zero value in this field indicates the presence of a debug port. The value in this register must not be greater than N_PORTS.

Bit 16 – P_INDICATOR Port Indicators
This bit indicates whether the ports support port indicator control. When this bit is a 1, the port status and control registers include a read/writeable field for controlling the state of the port indicator. See UHPHS Port Status and Control Register for a definition of the port indicator control field.

Bits 15:12 – N_CC[3:0] Number of Companion Controllers
This field indicates the number of companion controllers associated with this USB 2.0 host controller.
A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports.
A value larger than zero in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports.

Bits 11:8 – N_PCC[3:0] Number of Ports per Companion Controller
This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software.
For example, if N_PORTS has a value of 6 and N_CC has a value of 2, then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first four are routed to companion controller 1 and the last two are routed to companion controller 2.
The number in this field must be consistent with N_PORTS and N_CC.

Bit 4 – PPC Port Power Control
This field indicates whether the host controller implementation includes port power control. A one in this bit indicates the ports have port power switches. A zero in this bit indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register (see UHPHS Port Status and Control Register).

Bits 3:0 – N_PORTS[3:0] Number of Ports
This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1 to 15.
A zero in this field is undefined.

73.7.3 UHPHS Host Controller Capability Parameters Register

Name: UHPHS_HCCPARAMS
Offset: 0x08
Reset: 0x00000026
Property: Read-only

These fields define capability parameters: Multiple Mode control (time-base bit functionality), addressing capability, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EECP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IST[3:0]					ASPC	PFLF	AC
Access	R	R	R	R		R	R	R
Reset	0	0	1	0		1	1	0

Bits 15:8 – EECP[7:0] EHCI Extended Capabilities Pointer
Indicates the existence of a capabilities list. A value of 0 indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in the PCI configuration space of the first EHCI extended capability. The pointer value must be 64 or greater if implemented to maintain the consistency of the PCI header defined for this class of device.

Bits 7:4 – IST[3:0] Isochronous Scheduling Threshold
Indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is 0, the value of the least significant three bits indicates the number of microframes a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is set to 1, the host software assumes the host controller may cache an isochronous data structure for an entire frame.

Bit 2 – ASPC Asynchronous Schedule Park Capability
The park capability can be disabled or enabled and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the UHPHS_USBCMD register.

Value	Description
0	Host controller does not supports the park feature for high-speed queue.
1	Host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule.

Bit 1 – PFLF Programmable Frame List Flag

Value	Description
0	System software must use a frame list length of 1024 elements with this host controller. The UHPHS_USBCMD register Frame List Size field is a read-only register and must be set to 0.
1	System software can specify and use a smaller frame list and configure the host controller via the UHPHS_USBCMD register Frame List Size field. The frame list must always be aligned on a 4-Kbyte page boundary. This requirement ensures that the frame list is always physically contiguous.

Bit 0 – AC 64-bit Addressing Capability
This field documents the addressing range capability of this implementation. The value of this field determines whether software should use 32-bit or 64-bit data structures.
This information is not tightly coupled with the UHPHS_USBBASE address register mapping control. The 64-bit Addressing Capability bit indicates whether the host controller can generate 64-bit addresses as a host. The UHPHS_USBBASE register indicates the host controller only needs to decode 32-bit addresses as a client.

Value	Description
0	Data structures using 32-bit address memory pointers
1	Data structures using 64-bit address memory pointers

73.7.4 UHPHS USB Command Register

Name: UHPHS_USBCMD
Offset: 0x10
Reset: 0x00080B00
Property: Read/Write

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ITC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
					ASPME		ASPMC[1:0]	
Access					R-R/W		R-R/W	R-R/W
Reset					1		1	1

Bit	7	6	5	4	3	2	1	0
	LHCR	IAAD	ASE	PSE	FLS[1:0]		HCRESET	RS
Access	R/W	R/W	R/W	R/W	R-R/W	R-R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ITC[7:0] Interrupt Threshold Control
This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.

Value	Maximum Interrupt Interval
0	Reserved
1	1 microframe
2	2 microframes
4	4 microframes
8	8 microframes (1 ms)
16	16 microframes (2 ms)
32	32 microframes (4 ms)
64	64 microframes (8 ms)

Any other value in this register yields undefined results.
Software modifications to this field while HCHLT=0 results in undefined behavior.

Bit 11 – ASPME Asynchronous Schedule Park Mode Enable (optional)
If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this bit is set to 1 and is Read/Write. Otherwise the bit must be 0 and is read-only.

Value	Description
0	Park mode is enabled.
1	Park mode is disabled.

Bits 9:8 – ASPMC[1:0] Asynchronous Schedule Park Mode Count (optional)
If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this field defaults to 3 and is read/write. Otherwise it defaults to 0 and is read-only. It contains a count of the number of successive

transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1 to 3. Software must not write a 0 to this bit when Park Mode Enable is set to 1 as this will result in undefined behavior.

Bit 7 – LHCR Light Host Controller Reset (optional)
This control bit is not required. If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers. For example, the UHPHS_PORTSC registers should not be reset to their default values and the CF bit setting should not go to 0 (retaining port ownership relationships).
A host software read of this bit as 0 indicates the Light Host Controller Reset has completed and it is safe for host software to re-initialize the host controller. A host software read of this bit as 1 indicates the Light Host Controller Reset has not yet completed.
If not implemented, a read of this field will always return a 0.

Bit 6 – IAAD Interrupt on Async Advance Doorbell
This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell.
When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the UHPHS_USBSTS register. If the Interrupt on Async Advance Enable bit in the UHPHS_USBINTR register is set to 1, then the host controller will assert an interrupt at the next interrupt threshold.
The host controller sets this bit to 0 after it has set the Interrupt on Async Advance status bit in the UHPHS_USBSTS register to 1.
Software should not write a 1 to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.

Bit 5 – ASE Asynchronous Schedule Enable
This bit controls whether the host controller skips processing the Asynchronous Schedule.

Value	Description
0	Do not process the Asynchronous Schedule.
1	Use the UHPHS_ASYNCLISTADDR register to access the Asynchronous Schedule.

Bit 4 – PSE Periodic Schedule Enable
This bit controls whether the host controller skips processing the Periodic Schedule.

Value	Description
0	Do not process the Periodic Schedule.
1	Use the UHPHS_PERIODICLISTBASE register to access the Periodic Schedule.

Bits 3:2 – FLS[1:0] Frame List Size
This field is read-only with one exception: it is read/write if the Programmable Frame List flag, in the UHPHS_HCCPARAMS register, is set to 1. This field specifies the size of the frame list. The size of the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index.

Value	Description
0	1024 elements (4096 bytes).
1	512 elements (2048 bytes).
2	256 elements (1024 bytes), for resource-constrained environments.
3	Reserved.

Bit 1 – HCRESET Host Controller Reset
This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.
When software writes a 1 to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.
PCI Configuration registers are not affected by this reset. All operational registers, including port registers and port state machines, are set to their initial values. Port ownership reverts to the companion host controller(s) with side effects. Software must reinitialize the host controller in order to return the host controller to an operational state.
This bit is set to 0 by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a 0 to this register.

Software must not set this bit to 1 when HCHLT in the UHPHS_USBSTS register is 0. Attempting to reset an actively running host controller results in undefined behavior.

Bit 0 – RS Run/Stop
The Host Controller must halt within 16 microframes after software clears the bit RS. The HCHLT bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write 1 to this field unless the host controller is in the halted state (i.e., HCHLT in the UHPHS_USBSTS register is 1). Doing so yields undefined results.

Value	Description
0	Host Controller completes the current and any actively pipelined transactions on the USB and then halts.
1	Host Controller proceeds with execution of the schedule.

73.7.5 UHPHS USB Status Register

Name: UHPHS_USBSTS
Offset: 0x14
Reset: 0x00001000
Property: Read/Write

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ASS	PSS	RCM	HCHLT				
Access	R	R	R	R				
Reset	0	0	0	1				
Bit	7	6	5	4	3	2	1	0
			IAA	HSE	FLR	PCD	USBERRINT	USBINT
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 15 – ASS Asynchronous Schedule Status
The bit reports the current real status of the Asynchronous Schedule.
The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the UHPHS_USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled or disabled.

Value	Description
0	Asynchronous Schedule is disabled.
1	Asynchronous Schedule is enabled.

Bit 14 – PSS Periodic Schedule Status
The bit reports the current real status of the Periodic Schedule. If this bit is set to 0, then the status of the Periodic Schedule is disabled. If this bit is set to 1, then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the UHPHS_USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled or disabled.

Bit 13 – RCM Reclamation
This is a read-only status bit used to detect any empty asynchronous schedule.

Bit 12 – HCHLT HCHalted
This bit is 0 whenever the Run/Stop bit is 1. The Host Controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (following an internal error, for example).

Bit 5 – IAA Interrupt on Async Advance (Cleared on write)
System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing 1 to the Interrupt on the Async Advance Doorbell bit in the UHPHS_USBCMD register. This status bit indicates the assertion of that interrupt source.

Bit 4 – HSE Host System Error (Cleared on write)
The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Host Abort, and PCI Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.

Bit 3 – FLR Frame List Rollover (Cleared on write)
The Host Controller sets this bit to 1 when the Frame List Index (see UHPHS USB Frame Index Register) rolls over from its maximum value to 0. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the Frame List Size field of the UHPHS_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to 1 every time FRINDEX[12] toggles.

Bit 2 – PCD Port Change Detect (Cleared on write)
The Host Controller sets this bit to 1 when any port for which the Port Owner bit is set to 0 (see UHPHS Port Status and Control Register) has a change bit transition from 0 to 1 or a Force Port Resume bit transition from 0 to 1 as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Change being set to 1 after system software has relinquished ownership of a connected port by writing 1 to a port's Port Owner bit.
This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force Port Resume, Overcurrent Change, Enable/Disable Change and Connect Status Change).

Bit 1 – USBERRINT USB Error Interrupt (Cleared on write)
The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (error counter underflow, for example). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set.

Bit 0 – USBINT USB Interrupt (Cleared on write)
The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set.
The Host Controller also sets this bit to 1 when a short packet is detected (the actual number of bytes received was less than the expected number of bytes).

73.7.6 UHPHS USB Interrupt Enable Register

Name: UHPHS_USBINTR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Interrupt sources that are disabled in this register still appear in the UHPHS_USBSTS to allow the software to poll for events.

For all bits, 1=Enabled, 0=Disabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			IAAE	HSEE	FLRE	PCIE	USBEIE	USBIE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – IAAE Interrupt on Async Advance Enable
The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit UHPHS_USBSTS.

Bit 4 – HSEE Host System Error Enable
The interrupt is acknowledged by software clearing the Host System Error bit in UHPHS_USBSTS.

Bit 3 – FLRE Frame List Rollover Enable
The interrupt is acknowledged by software clearing the Frame List Rollover in UHPHS_USBSTS.

Bit 2 – PCIE Port Change Interrupt Enable
The interrupt is acknowledged by software clearing the Port Change Detect bit in UHPHS_USBSTS.

Bit 1 – USBEIE USB Error Interrupt Enable
The interrupt is acknowledged by software clearing the USBERRINT in UHPHS_USBSTS.

Bit 0 – USBIE USB Interrupt Enable
The interrupt is acknowledged by software clearing the USBINT in UHPHS_USBSTS.

73.7.7 UHPHS USB Frame Index Register

Name: UHPHS_FRINDEX
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register is used by the host controller to index into the periodic frame list. The register updates every 125 μs (once each microframe). Bits [N:3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the UHPHS_USBCMD register).

This register must be written as a DWord. Byte writes produce undefined results. This register cannot be written unless the Host Controller is in the Halted state as indicated by the HCHLT bit (UHPHS_USBSTS register). A write to this register while the Run/Stop bit is set to 1 (UHPHS_USBCMD register) produces undefined results. Writes to this register also affect the SOF value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			FI[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FI[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – FI[13:0] Frame Index
The value in this register increments at the end of each time frame (microframe, for example). Bits [N:3] are used for the Frame List current index. This means that each location of the frame list is accessed eight times (frames or microframes) before moving to the next index. The following illustrates values of N based on the value of FLS (Frame List Size) in the UHPHS_USBCMD register.

UHPHS_USBCMD.FLS	Number Elements	N
0	1024	12
1	512	11
2	256	10
3	Reserved	–

The SOF frame number value for the bus SOF token is derived or alternatively managed from this register. The value of FRINDEX must be 125 μs (1 microframe) ahead of the SOF token value. The SOF value may be implemented as an 11-bit shadow register. For this discussion, this shadow register is 11 bits and is named SOFV. SOFV updates every eight microframes (1 millisecond). An example implementation to achieve this behavior is to increment SOFV each time the FRINDEX[2:0] increments from 0 to 1.
Software must use the value of FRINDEX to derive the current microframe number, both for high-speed isochronous scheduling purposes and to provide the “get microframe number” function required for client drivers. Therefore, the value of FRINDEX and the value of SOFV must be kept consistent if chip is reset or software writes to FRINDEX.

Writes to FRINDEX must also write-through FRINDEX[13:3] to SOFV[10:0]. In order to keep the update as simple as possible, software should never write a FRINDEX value where the three least significant bits are 7 or 0.

73.7.8 UHPHS Periodic Frame List Base Address Register

Name: UHPHS_PERIODICLISTBASE
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (UHPHS_FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence. This register must be written as a DWord. Byte writes produce undefined results.

Bit	31	30	29	28	27	26	25	24
	BA[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BA[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BA[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:12 – BA[19:0] Base Address (Low)
These bits correspond to memory address signals [31:12], respectively.

73.7.9 UHPHS Asynchronous List Address Register

Name: UHPHS_ASYNCLISTADDR
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

This 32-bit register contains the address of the next asynchronous queue head to be executed. Bits [4:0] of this register cannot be modified by system software and will always return a zero when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte (cache line) aligned. This register must be written as a DWord. Byte writes produce undefined results.

Bit	31	30	29	28	27	26	25	24
	LPL[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPL[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LPL[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPL[2:0]							
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bits 31:5 – LPL[26:0] Link Pointer Low
These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).

73.7.10 UHPHS Configure Flag Register

Name: UHPHS_CONFIGFLAG
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CF
Access								R/W
Reset								0

Bit 0 – CF Configure Flag
Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below.

Value	Description
0	Port routing control logic default-routes each port to an implementation-dependent classic host controller (default value).
1	Port routing control logic default-routes all ports to this host controller.

73.7.11 UHPHS Port Status and Control Register

Name: UHPHS_PORTSCx
Offset: 0x54 + x*0x04 [x=0..2]
Reset: 0x00003000
Property: Read/Write

The number of port registers is documented in the UHPHS_HCSPARAMS register. Software uses this information as an input parameter to determine how many ports need to be serviced. All ports have the structure defined below.

This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port has port power control, software cannot change the state of the port until after it applies power to the port by setting port power to a 1. Software must not attempt to change the state of the port until after power is stable on the port. The host is required to have power stable to the port within 20 milliseconds of the 0 to 1 transition.

Notes:
1. When a device is attached, the port state transitions to the connected state and system software will process this as with any status change notification.
2. If a port is being used as the Debug Port, then the port may report device connected and enabled when the Configured Flag is set to 0.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PIC[1:0]		PO	PP	LS[1:0]			PR
Access	R/W	R/W	R/W	R-R/W	R	R		R/W
Reset	0	0	1	1	0	0		0

Bit	7	6	5	4	3	2	1	0
	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
Access	R/W	R/W	R/W	R	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit 22 – WKOC_E Wake on Overcurrent Enable
This field is 0 if Port Power is 0.

Value	Description
`0`	Disables the port to be sensitive to overcurrent conditions as wake-up events.
`1`	Enables the port to be sensitive to overcurrent conditions as wake-up events.

Bit 21 – WKDSCNNT_E Wake on Disconnect Enable
This field is 0 if Port Power is 0.

Value	Description
`0`	Disables the port to be sensitive to device disconnects as wake-up events.
`1`	Enables the port to be sensitive to device disconnects as wake-up events.

Bit 20 – WKCNNT_E Wake on Connect Enable
This field is 0 if Port Power is 0.

Value	Description
0	Disables the port to be sensitive to device connects as wake-up events.
1	Enables the port to be sensitive to device connects as wake-up events.

Bits 19:16 – PTC[3:0] Port Test Control
When this field is set to 0, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value.
Test mode bits are encoded as follows (6 to 15 are reserved):

Value	Test Mode
0	Test mode not enabled
1	Test J_STATE
2	Test K_STATE
3	Test SE0_NAK
4	Test Packet
5	Test FORCE_ENABLE

Refer to the USB Specification Revision 2.0, Chapter 7, for details on each test mode.

Bits 15:14 – PIC[1:0] Port Indicator Control
Writing to these bits has no effect if the P_INDICATOR bit in the UHPHS_HCSPARAMS register is set to 0. If the P_INDICATOR bit is set to 1, then the bits are encoded as follows:

Value	Meaning
0	Port indicators are off
1	Amber
2	Green
3	Undefined

Refer to the USB Specification Revision 2.0 for a description of how these bits are to be used.
This field is 0 if Port Power is 0.

Bit 13 – PO Port Owner
System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes 1 to this bit when the attached device is not a high-speed device. A 1 in this bit means that a companion host controller owns and controls the port.

Value	Description
0	This bit unconditionally goes to a 0 when the bit UHPHS_CONFIGFLAG.CF makes a 0 to 1 transition.
1	This bit unconditionally goes to 1 whenever the bit UHPHS_CONFIGFLAG.CF=0.

Bit 12 – PP Port Power
The function of this bit depends on the value of the Port Power Control (PPC) field in the UHPHS_HCSPARAMS register. When host controller has port power control switches (PPC=0), PP is in read-only mode:

Value	Description
1	Each port is hard-wired to power.

When host controller has port power control switches (PPC=1), PP is in read/write mode:

Value	Description
0	Host port power switch is off. When power is not available on a port (i.e., PP at 0), the port is non-functional and does not report attaches, detaches, etc.

...........continued	
Value	**Description**
1	Host port power switch is on. When power is not available on a port (i.e., PP at 0), the port is non-functional and does not report attaches, detaches, etc.

When an overcurrent condition is detected on a powered port and PPC is set to 1, the PP bit in each affected port may be transitioned by the host controller from 1 to 0 (removing power from the port).

Bits 11:10 – LS[1:0] Line Status
These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is 0 and the current connect status bit is set to 1.
This value of this field is undefined if Port Power is 0.

Value	Name	Description
0	SE0	Not a low-speed device, perform EHCI reset
1	K-STATE	Low-speed device, release ownership of port
2	J-STATE	Not a low-speed device, perform EHCI reset
3	Undefined	Not a low-speed device, perform EHCI reset

Bit 8 – PR Port Reset
When software writes a 1 to this bit (from 0), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a 0 to this bit to terminate the bus reset sequence. Software must keep this bit set to 1 long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes.
Note: When software writes this bit to 1, it must also write 0 to the Port Enable bit.

When software writes a 0 to this bit, there may be a delay before the bit status changes to 0. The bit status will not read as 0 until after the reset has completed. If the port is in High-Speed mode after reset is complete, the host controller will automatically enable this port (set the Port Enable bit to 1, for example). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from 1 to 0. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2 ms of software writing this bit to 0.
The HCHLT bit in the UHPHS_USBSTS register should be set to 0 before software attempts to use this bit. The host controller may hold Port Reset asserted to 1 when the HCHLT bit is 1.
This field is 0 if Port Power is 0.

Value	Description
0	Port is not in Reset.
1	Port is in Reset.

Bit 7 – SUS Suspend

Value	Description
0	Port not in Suspend state.
1	Port in Suspend state.

Note:
Port Enabled Bit and Suspend bit of this register define the port states as follows:

Bits [Port Enabled, Suspend]	Port State
0X	Disable
10	Enable
11	Suspend

When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.

A write of 0 to this bit is ignored by the host controller. The host controller will unconditionally set this bit to 0 when:
- Software sets the Force Port Resume bit to 0 (from 1).
- Software sets the Port Reset bit to 1 (from 0).

If host software sets this bit to 1 when the port is not enabled (i.e., Port Enabled bit set to 0), the results are undefined.

This field is 0 if Port Power is set to 0.

Bit 6 – FPR Force Port Resume
This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended (Suspend and Enabled bits are set to 1) and software transitions this bit to 1, then the effects on the bus are undefined.
Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to 1 because a J-to-K transition is detected, the Port Change Detect bit in the UHPHS_USBSTS register is also set to 1. If software sets this bit to 1, the host controller must not set the Port Change Detect bit.
Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains set to 1. Software must appropriately time the Resume and set this bit to 0 when the appropriate amount of time has elapsed. Writing a 0 (from 1) causes the port to return to High-Speed mode (forcing the bus below the port into a high-speed idle). This bit will remain set to 1 until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to 0.
This field is 0 if Port Power is 0.

Value	Description
0	No resume (K-state) detected/driven on port.
1	Resume detected/driven on port.

Bit 5 – OCC Overcurrent Change (Cleared on write)
Software clears this bit by writing 1.

Value	Description
0	No change to Overcurrent Active.
1	Changes to Overcurrent Active.

Bit 4 – OCA Overcurrent Active
This bit will automatically transition from 1 to 0 when the overcurrent condition is removed.

Value	Description
0	This port does not have an overcurrent condition.
1	This port currently has an overcurrent condition.

Bit 3 – PEDC Port Enable/Disable Change (Cleared on write)
For the root hub, this bit gets set to 1 only when a port is disabled due to the appropriate conditions existing at the EOF2 point (refer to Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.
This field is 0 if Port Power bit is 0.

Value	Description
0	No change in port enabled/disabled status.
1	Port enabled/disabled status has changed.

Bit 2 – PED Port Enabled/Disabled
Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this field. The host controller will only set this bit to 1 when the reset sequence determines that the attached device is a high-speed device.
Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.
When the port is disabled (0b), downstream propagation of data is blocked on this port, except for reset.
This field is 0 if Port Power bit is 0.

Value	Description
0	Disable.
1	Enable.

Bit 1 – CSC Connect Status Change (Cleared on write)
Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change.
For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be “setting” an already-set bit (i.e., the bit remains set). Software sets this bit to 0 by writing a 1 to it.
This field is 0 if Port Power bit is 0.

Value	Description
0	No change.
1	Change in Current Connect Status.

Bit 0 – CCS Current Connect Status
This value reflects the current state of the port, and may not correspond directly to the event that caused the CSC bit to be 1.
This bit is 0 if Port Power is 0.

Value	Description
0	No device is present.
1	Device is present on port.

73.7.12 EHCI: REG06 - AHB Error Status

Name: UHPHS_INSNREG06
Offset: 0xA8
Reset: 0x00000000
Property: Read/Write

Control and Status Register, used to read the UTMI registers from the signals below.

Bit	31	30	29	28	27	26	25	24
	AHB_ERR							
Access	R/W							
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					HBURST[2:0]			Nb_Burst[4]
Access					R	R	R	R
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Nb_Burst[3:0]				Nb_Success_Burst[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – AHB_ERR AHB Error
System bus error was encountered and erroneous burst characteristics are captured. To clear this field the application must write a 0.
EHCI:

- When no error, 0 is written to INSNREG06[8:4].
- When INCR4 and an error occurs, 4 is written to INSNREG06[8:4].
- When INCR8 and an error occurs, 8 is written to INSNREG06[8:4].
- When INCR16 and an error occurs, 16 is written to INSNREG06[8:4].
- Other values except 4, 8, and 16 are not written to INSNREG06[8:4].

OHCI:

- When no error, 0 is written to INSNREG06[8:4].
- When INCR4 and error occurs, 4 is written to INSNREG06[8:4].
- Other values except 4 are not written to INSNREG06[8:4].

Bits 11:9 – HBURST[2:0] Burst Value
Value of the control phase at which the AHB error occurred.
This field applies to enabled incremental bursts only.

Bits 8:4 – Nb_Burst[4:0] Number of Bursts
Number of beats expected in the burst at which the AHB error occurred. Valid values are 0 to 16.
This field applies to enabled incremental bursts only.

Bits 3:0 – Nb_Success_Burst[3:0] Number of Successful Bursts
Number of successfully completed beats in the current burst before the AHB error occurred.
This field applies to enabled incremental bursts only.

73.7.13 EHCI: REG07 - AHB Host Error Address

Name: UHPHS_INSNREG07
Offset: 0xAC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	AHB_ADDR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	AHB_ADDR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	AHB_ADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	AHB_ADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AHB_ADDR[31:0] AHB Address
System bus address of the control phase at which the system bus error occurred.

74. Electrical Characteristics

74.1 Electrical Parameters Usage

Tables in the following sections define the limiting values for several electrical parameters: 74.5. I/O Characteristics, 74.6. Embedded Analog Peripherals Characteristics, 74.7. Security Module Characteristics 74.8. Power Consumption in Active Mode, 74.9. Operation and Power Consumption in Low-Power Modes.

- Unless otherwise noted, these values are valid over the junction temperature range defined in Ordering Information. This junction temperature range is referred to as the “applicable T_J range" in the following sections.
- Parameters annotated as "Simulation data" are not production-tested. Their limiting values come from simulations run in corner case conditions and were verified by electrical characterization over a limited number of samples.
- These limits may be affected by the board on which the device is mounted. In particular, noisy supply and ground conditions must be avoided and care must be taken to provide:
  – a PCB with a low-impedance ground plane. A single unbroken ground plane is a minimum requirement.
  – low-impedance decoupling of the device power supply inputs. A 10 nF to 220 nF Ceramic X7R (or X5R) capacitor placed very close to each power supply input is a minimum requirement. See specific recommendations regarding analog pins or functions in the corresponding sections. To reduce any potential electromagnetic compatibility (EMC) related issues, it is good practice to double this decoupling capacitor whenever possible with a high frequency one, for example one 100 pF (C0G or NP0) per power supply input.
  – low impedance power supply decoupling of external components. This recommendation aims at avoiding large current spikes flowing into the PCB ground and power planes.
- In addition, although the device is specified with wide operating supply ranges on most of its supply inputs (for example 1.7V to 3.6V), large and fast supply variations may lead to unpredictable device behavior including, but not limited to, out-of-specification operation. Therefore, in addition to maintaining the power supply inputs within their specified range, it is also mandatory to keep the power supply variations within the limits described in the Maximum Power Supply Variations table during the device operation.
- Finally, the device performances and operating junction temperature are strongly dependent on the thermal performances of the board on which the device mounted.

Table 74-1. Maximum Power Supply Variations(1)

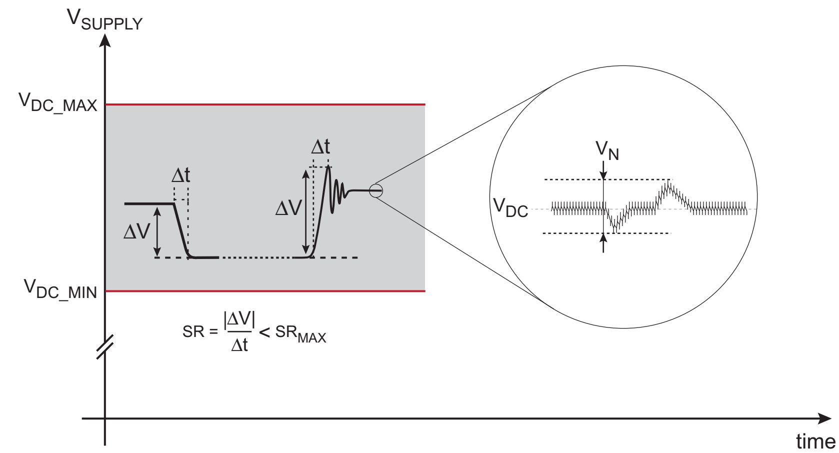
Symbol	Parameter	Conditions	Min	Max	Unit
V_N	Peak-to-peak ripple and noise voltage	Applies to VDDCORE, VDDCPU, VDDIOP0, VDDIOP1, VDDQSPI0, VDDQSPI1, VDDIODDR, VDDSDMMC0, VDDSDMMC1, VDDSDMMC2	–	3	% V_{DC}(2)
		Applies to VDDIN33, VBAT, VDDUTMII	–	1	% V_{DC}(2)
SR	Slewrate of power supply variations	$\Delta V \leq 5\% \ V_{DC\_MIN}$(3)(4)	–	±50	V/ms
		$\Delta V < 10\% \ V_{DC\_MIN}$	–	±10	V/ms
		$\Delta V \geq 10\% \ V_{DC\_MIN}$	–	±1	V/ms

Notes:
1. VDDANA and VDDDHY are not mentioned in this table, as they must be connected to the VDDOUT25 regulator output that fulfills the electrical requirements of these power inputs.
2. V_{DC} is the DC value of the power supply.
3. V_{DC_MIN} is the minimum operating voltage of the supply input as described in the table SAMA7G5 Power Supply Inputs.
4. ΔV is the amplitude of the variation. The slew rate specification applies when the $\Delta V \geq V_N$.

The following examples and figure illustrate this table:

- When working with VDDIOP0 = 3.3V, a maximum power supply ripple and noise voltage of 99 mV peak-to-peak (3% of 3.3V) must be respected.
- When working with VDDIN33 = 3.3V, a maximum power supply ripple and noise voltage of 33 mV peak-to-peak (1% of 3.3V) must be respected.

Figure 74-1. Maximum Power Supply Variation



74.2 Absolute Maximum Ratings

Table 74-2. Absolute Maximum Ratings

Storage Temperature	-60°C to +150°C	**Note:** Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. **Exposure to absolute maximum rating conditions for extended periods may affect device reliability.**
Voltage Difference between two ground pins (among GND, GNDANA and GNDIN33)	±50 mV	
Power Supply Inputs with respect to ground pins:		
VDDCORE, VDDCPU	-0.3V to 1.4V	
VDDIODDR	-0.3V to 2.0V	
VDDIOPx, VDDQSPIx, VDDSDMMCx	-0.3V to 4.0V	
VDDIN33, VBAT	-0.3V to 4.0V	
VDDUTMII	-0.3V to 4.0V	
VDDDPHY, VDDANA	-0.3V to 3.0V	
Voltage on Digital or Analog Input Pins with respect to ground	-0.3V to 4.0V	
Injected Current into any input pin	± 1 mA	
Total Injected Current in all input pins of a common power supply pair	± 10 mA	
Maximum DC Output Current:		
On all I/O lines into one power rail	100 mA	
Per output pin	25 mA	

Note: All I/O pins are internally clamped to their respective VDD and GND rails, as defined in the table Pin Description. Ex: for PA12, this corresponds to VDDIOP0 and GND.

74.3 Recommended Operating Conditions

Table 74-3. Power Supply Inputs

Power Input	Parameter	Conditions	Min	Max	Unit
VDDCPU[(1)]	Cortex-A7 and cache memories	$f_{CPU\_CLK} \leq$ 600 MHz	1.03	1.21	V
		$f_{CPU\_CLK} \leq$ 800 MHz	1.12	1.21	V
		$f_{CPU\_CLK} \leq$ 1 GHz[(2)]	1.22	1.30	V
VDDCORE	Core logic power supply	–	1.12	1.21	V
VDDIODDR	SDRAM I/O lines power supply	[LPDDR2 / LPDDR3]-SDRAM	1.14	1.30	V
		DDR3-SDRAM	1.425	1.575	V
		DDR3L-SDRAM	1.283	1.45	V
		DDR2-SDRAM	1.7	1.9	V
VDDIN33[(3)]	VDDOUT25 regulator, backup power switch and OTP power inputs	–	3.0	3.6	V
VDDUTMII[(3)]	USB device and host UTMI+ interface	–	3.0	3.6	V
VDDDPHY[(4)]	MIPI DPHY power supply	–	2.4	2.6	V
VDDANA[(4)]	ADC, comparator, temperature sensor, PLLs, main crystal oscillator, main RC oscillator power supply	–	2.4	2.6	V
VDDIOP[0,1][(5)]	Peripheral I/O lines	–	1.7	3.6	V
VDDQSPI[0,1][(5)]	QSPIx I/O lines	–	1.7	3.6	V
VDDSDMMC[0,1,2][(5)]	SDMMCx I/O lines	–	1.7	3.6	V
VBAT	Backup supply input	–	1.7	3.6	V
t_{R_VDD}	Power supply slope at power-up	Applies to any of the power supply inputs listed above	0.2	20	mV/µs
t_{F_VDD}	Power supply slope at power-down		-20	-1[(6)]	mV/µs

Notes:
1. For lifetime estimation as a function of operating voltage and junction temperature, contact your local sales representative.
2. VDDCPU related alarms in SECUMOD_SR are always triggered when f_{CPU_CLK} > 800 MHz
3. VDDIN33 and VDDUTMII are powered from one single power source so that ΔV(VDDIN33,VDDUTMII) ≤ 50 mV.
4. VDDANA and VDDDPHY must be connected to VDDOUT25.
5. Supply range restrictions apply when using the digital peripheral timing characteristics. See I/O Characteristics.
6. For VBAT, this value is 0 mV/µs.

Table 74-4. Recommended Operating Conditions on Input Pins[(1)]

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IN}	Input line voltage range on inputs[(2)(3)]	–	-0.3	V_{DD} + 0.3	V
I_{IN}	DC current injection on inputs[(4)(5)]	–	–	± 0.2	mA

...........continued					
Symbol	Parameter	Conditions	Min	Max	Unit
I_{TOT_INJ}	Total current injection per power rail or per ground rail(6)	–	–	± 2	mA

Notes:

1. In this table, VDD refers to the voltage of the associated power rail of the I/O line, as defined in the table Pin Description. Ex: for PA12, VDD refers to VDDIOP0.
2. Input voltages $V_{IN} \leq 0V$ or $V_{IN} \geq VDD$ lead to negative or positive current injection on inputs.
3. For A/D converter analog inputs (PC13..PC24, PC30, PC31, PD0, PD1), input voltages $V_{IN} \geq \min(V_{DDANA}, V_{ADVREFP})$ lead to saturated A/D conversion to 0xFFF.
4. Current injection on A/D converter analog inputs (PC13..PC24, PC30, PC31, PD0, PD1) may degrade the analog performance of the corresponding channel, or the analog performance of other analog channels.
5. High frequency current injection must be limited to avoid propagating high frequency signals to internal sensitive analog circuits (oscillators, regulators, etc.). One common use case of high frequency current injection occurs when a digital input pin suffers overshoots and/or undershoots from a poorly adapted transmission line (PCB trace with signal reflections, for example). These cases should be cured by appropriate source series resistor termination. Special attention must be paid to high speed interfaces (Gigabit Ethernet MAC I/F, SD Card or e.MMC I/F, QSPI I/F, etc.).
6. Corresponds to the sum of the positive currents into one power rail and respectively to the sum of the negative currents into one ground rail as defined in the table Pin Description.

Table 74-5. Recommended Operating Conditions on Internal Clocks

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CPU_CLK}	Processor clock (CPU_CLK) frequency	$V_{DDCPU} \geq 1.22V$	–	1000	MHz
		$V_{DDCPU} \geq 1.12V$	–	800	MHz
		$V_{DDCPU} \geq 1.03V$	–	600	MHz
f_{MCK0}	Main system bus clock (MCK0) frequency	–	–	200	MHz
f_{MCK1}	Main system bus clock (MCK1) frequency	–	–	200	MHz
f_{MCK2}	Main system bus clock (MCK2) frequency	–	–	533	MHz
f_{MCK3}	Main system bus clock (MCK3) frequency	–	–	266	MHz
f_{MCK4}	Main system bus clock (MCK4) frequency	–	–	400	MHz

Table 74-6. Recommended Operating Conditions on SDRAM Interface

Symbol	Parameter	Conditions	Min	Max	Unit
f_{DDR_CLK}	SDRAM clock frequency	[LPDDR2 / LPDDR3]-SDRAM	100	533	MHz
		[DDR3 / DDR3L]-SDRAM	100	533	MHz
		DDR2-SDRAM	125	533	MHz

Table 74-7. Recommended Thermal Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
T_J	Junction temperature range	Ordering Code SAMA7G54-V/4HB	-40	105	°C

Table 74-8. BGA343 Package Thermal Characteristics[1][2]

Symbol	Parameter	Typ	Unit
R_{JA}	Junction-to-ambient thermal resistance	25	°C/W
R_{JB}	Junction-to-board thermal resistance	18	°C/W
R_{JC}	Junction-to-case thermal resistance	8	°C/W
Psi_{JT}	Junction-to-top characterization parameter	0.3	°C/W

Notes:
1. According to the JEDEC JESD51-2 standard, with 2s2p board and 0 m/s air flow.
2. These values are not directly applicable to the board where the device is mounted. As per JEDEC standards, these parameters do not characterize the package itself but rather the package together with the PCB (4-layer or more) and other environmental factors (still air, etc.). For example, in still-air JEDEC-defined R_{JA} measurements, almost 70%–95% of the power generated by the chip is dissipated from the test board, not from the surfaces of the package.

74.4 Recommended Power Supply Sequencing

In the following sections, various recommended power sequences are described. Operating the device outside this scope may lead to unpredictible behavior.

74.4.1 Power-Up and Power-Down

At power-up, from a power supply sequencing perspective, the SAMA7G5 power supply inputs are categorized into three independent groups:
- VDDIN33,
- the core group that contains VDDCPU and VDDCORE, and
- the periphery group that contains all other power supply inputs except VBAT.

The following figure shows the recommended power-up sequence. Note the following:
- VBAT
  - When supplied from a pre-charged storage element (battery or supercapacitor), VBAT is an always-on supply input and is therefore not part of the power supply sequencing.
  - When no storage element is used on VBAT in the application, VBAT must be tied to VDDIN33.
  - When a supercapacitor is used in the application to power VBAT during Backup mode, this element must be isolated from VBAT during its (slow) charge so that VBAT closely follows VDDIN33. In the table Power-Up Timing Specification, the parameter t_0 limits the delay to establish VBAT after VDDIN33.
- VDDOUT25 is the output of the internal VDDOUT25 regulator, therefore, there is no power supply requirement on this pin. VDDOUT25 is mentioned in the following figures for information only. VDDOUT25 is automatically started at VDDIN33 rise when VDDIN33 is above its power-on reset threshold.
- VDDDPHY and VDDANA must be connected to VDDOUT25 and are therefore not subject to any supply sequencing requirement.
- NRST must be asserted low during the whole power-up sequence.

Figure 74-2. Recommended Power Sequence at Power-Up

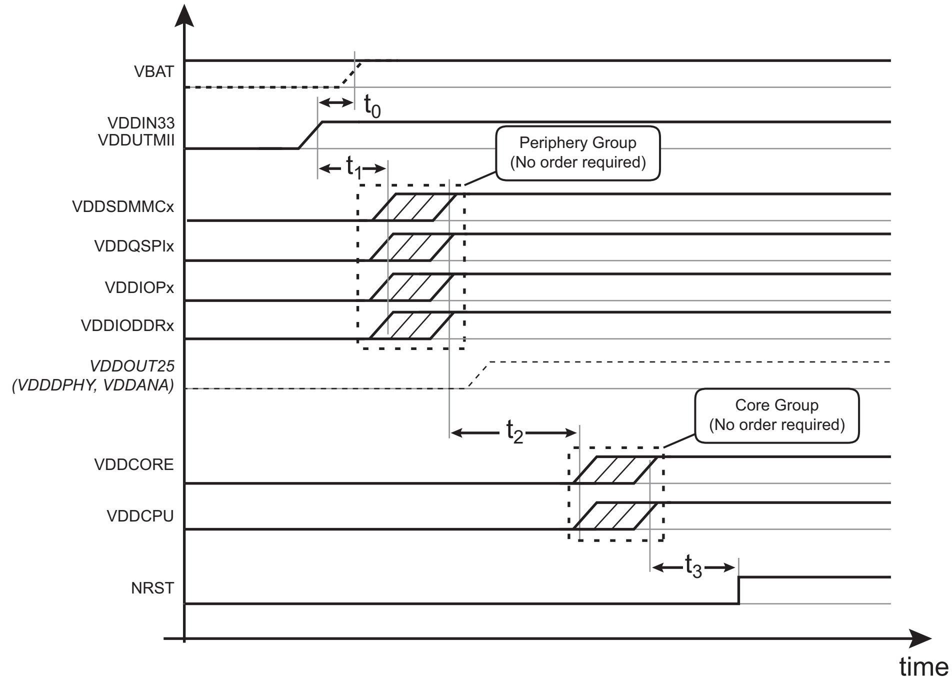


Table 74-9. Power-Up Timing Specification[(1)]

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VBAT delay	Delay from established VDDIN33 to established VBAT	–	0.2	ms
t_1	VDDIN33 to periphery group delay	Delay from established VDDIN33 to the first established periphery group supply	-0.1	–	ms
t_2	Periphery group to VDDCORE delay	Delay from the last established periphery group supply to the first core group supply turn-on	0	–	ms
t_3	Reset delay at power-up	Delay from the last established core group supply to NRST high	8	–	ms

Note:

1. The term "established" refers to a power supply established at 90% of its final value.

The following figure shows the SAMA7G5 power-down sequence that starts by asserting the NRST line to 0. Once NRST is asserted, the supply inputs can be immediately shut down without any specific timing or order. VBAT may not be shut down if the application uses a backup storage element on this supply input. When VDDIN33 falls below the negative-going threshold of the VDDIN33 power-on reset, the VDDOUT25 regulator is automatically shut down and its output is pulled low by an internal discharge resistor.

Figure 74-3. Recommended Power Sequence at Power-Down

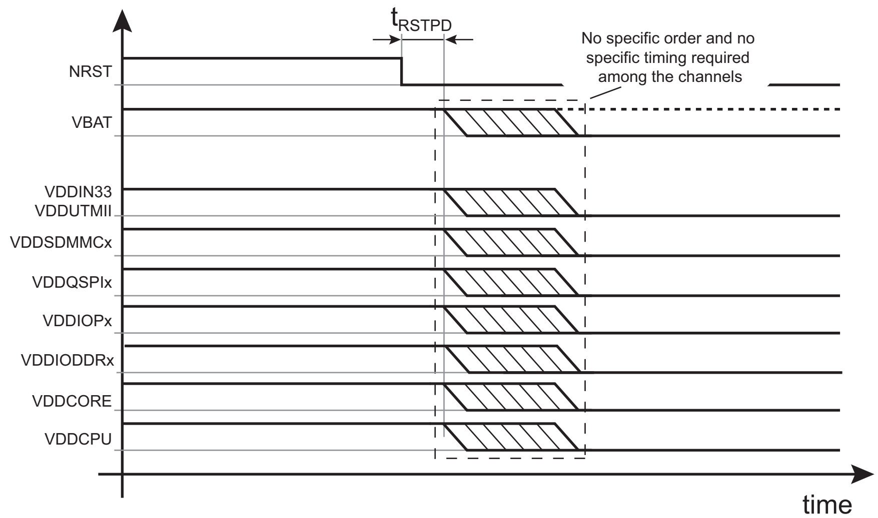


Table 74-10. Power-Down Timing Specification

Symbol	Parameter	Conditions	Min	Max	Unit
t_{RSTPD}	NRST Delay at power-down	Delay from NRST low to the first supply out of its operating range	0	–	ms

74.4.2 Backup Mode Entry and Wake-up

The following figure shows the recommended power-down sequence to place the device in Backup mode. The SHDN signal, output of the Shutdown Controller (SHDWC), signals the shutdown request to the external power supply. This output is supplied by VBAT present in Backup mode. In a similar way to the power-down sequence, the NRST signal must be asserted prior to turning the device power supplies off.

Figure 74-4. Recommended Power Sequence at Backup Mode Entry

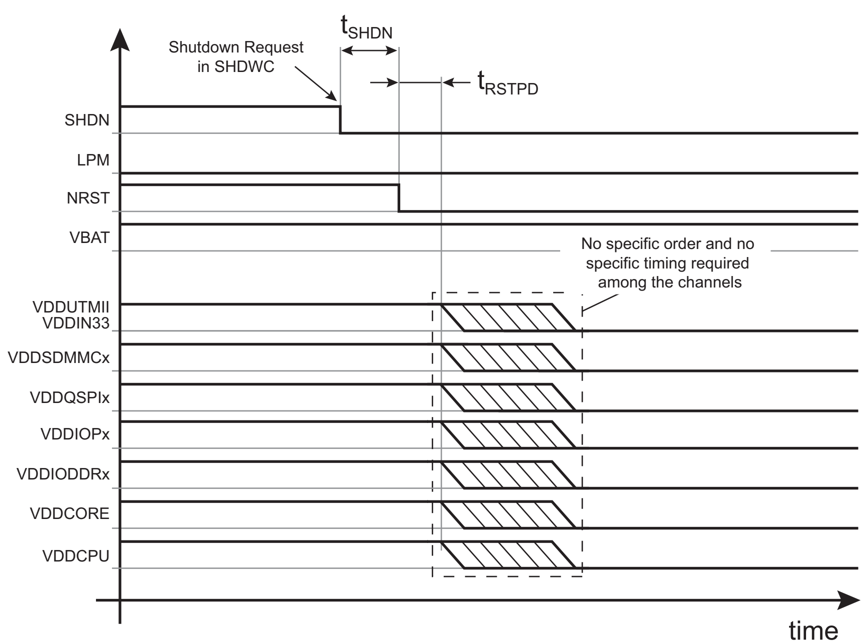


Table 74-11. Backup Mode Entry Timing Requirements

Symbol	Parameter	Conditions	Min	Max	Unit
t_{SHDN}	SHDN delay at backup mode entry	Delay from SHDN low to NRST low	0	–	ms
t_{RSTPD}	NRST delay at power-down	Delay from NRST low to the first supply out of its operating range	0	–	ms

The following figure shows the recommended power-up sequence to wake up the device from Backup mode. Upon a Wake-up event (WKUP pin, RTC alarm, etc.), the Shutdown Controller automatically toggles its SHDN output back to VBAT to request the external power supply to restart. This power-up sequence is the same as the one shown in the figure Recommended Power Sequence at Power-Up. In particular, the supply groups definition is the same.

Figure 74-5. Recommended Power Sequence at Wake-Up from Backup Mode

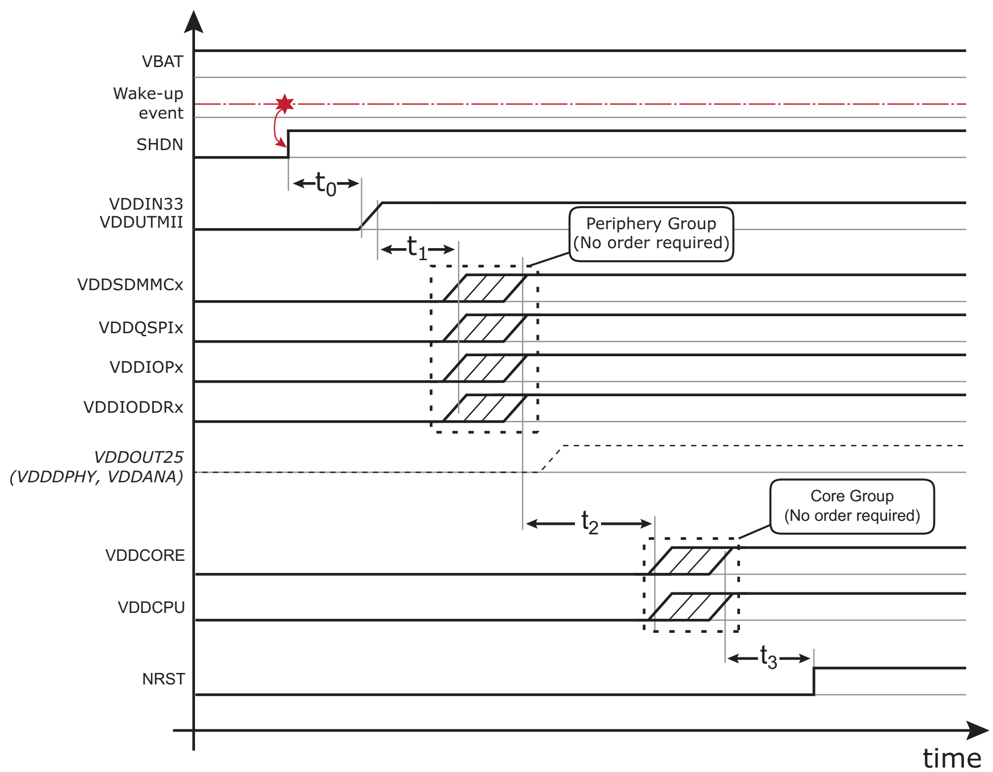


Table 74-12. Wake-Up from Backup Mode Timing Specification[(1)]

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VDDIN33 delay	Delay from SHDN high to VDDIN33 turn-on	0	–	ms
t_1	VDDIN33 to periphery group delay	Delay from established VDDIN33 to the first established periphery group supply	-0.1	–	ms
t_2	Periphery group to VDDCORE delay	Delay from the last periphery group established supply to VDDCORE supply turn-on	0	–	ms
t_3	Reset delay at power-up	From established VDDCORE to NRST high	8	–	ms

Note:

1. The term "established" refers to a power supply established at 90% of its final value.

74.4.3 BSR Mode Entry and Wake-up

The following figure shows the recommended power-down sequence to place the device in BSR mode (Backup Mode with SDRAM in Self-refresh mode to save the context). A combination of the LPM and SHDN signals, both outputs of the Shutdown Controller (SHDWC), requests a special powering mode to the external power supply where VDDIODDR is maintained. These outputs are supplied by VBAT present in BSR mode. In a similar way to the power-down sequence, the NRST signal must be asserted prior to turning the SAMA7G5 power supplies off.

Figure 74-6. Recommended Power Sequence at BSR Mode Entry

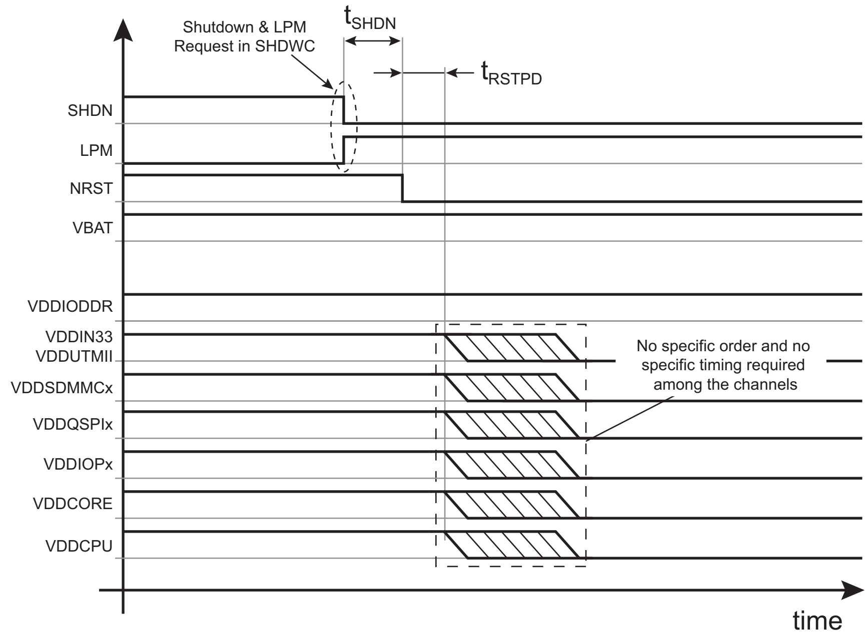


Table 74-13. BSR Mode Entry Timing Specification

Symbol	Parameter	Conditions	Min	Max	Unit
t_{SHDN}	SHDN delay at backup mode entry	Delay from SHDN low to NRST low	0	–	ms
t_{RSTPD}	NRST delay at power-down	Delay from NRST low to the first supply out of its operating range	0	–	ms

The following figure shows the recommended power-up sequence to wake up the device from BSR mode. Upon a Wake-up event (WKUP pin, RTC alarm, etc.), the Shutdown Controller automatically toggles its SHDN output back to VBAT to request the external power supply to restart. This power-up sequence is the same as the one shown in Recommended Power Sequence at Power-Up. In particular, the supply groups definition is the same. The LPM pin is not automatically reset, and therefore the external power supply may enter a power saving state at wake-up. To make the power supply return to normal operation, the LPM pin must be software-reset in the SHDWC as soon as possible in the wake-up process.

Figure 74-7. Recommended Power Sequence at Wake-Up from BSR Mode

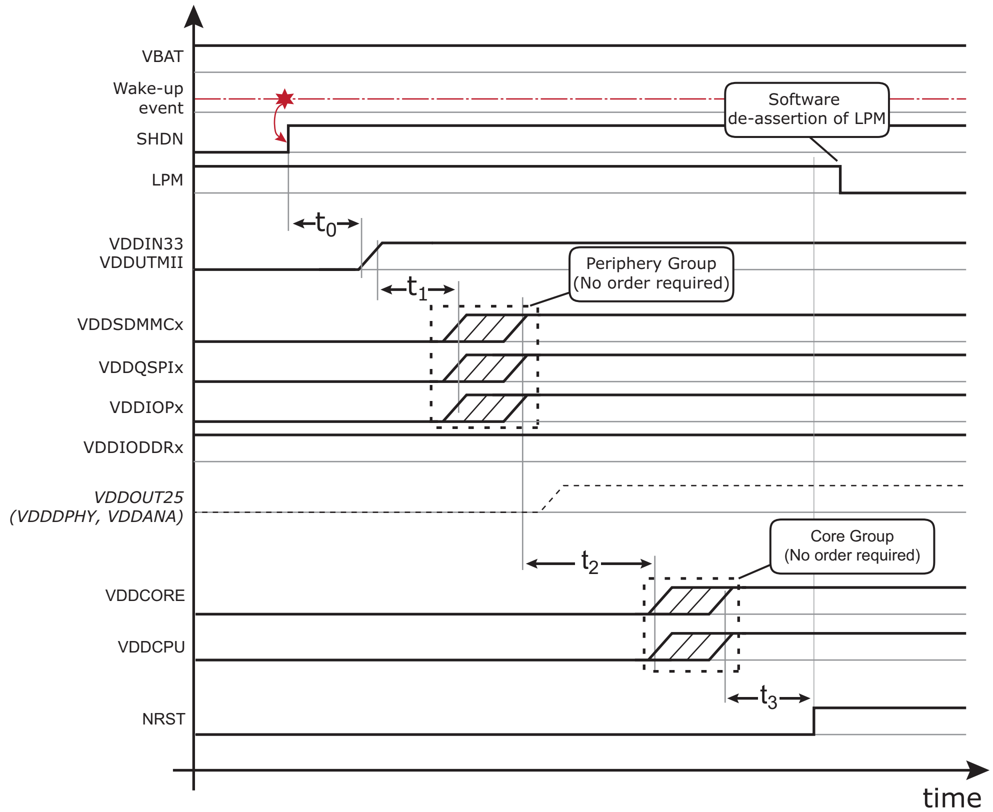


Table 74-14. Wake-Up from BSR Mode Timing Specification[(1)]

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VDDIN33 delay	Delay from SHDN high to VDDIN33 turn-on	0	–	ms
t_1	VDDIN33 to periphery group delay	Delay from established VDDIN33 to the first periphery group supply established	-0.1	–	ms
t_2	Periphery group to VDDCORE delay	Delay from the last periphery group established supply to VDDCORE supply turn-on	0	–	ms
t_3	Reset delay at power-up	From established VDDCORE to NRST high	8	–	ms

Note:

1. The term "established" refers to a power supply established at 90% of its final value.

74.5 I/O Characteristics

The device features the following types of input/output (I/O) circuits:

- GPIO I/Os with drive and slewrate control.
  - AUDIOCLK is a special case of GPIO type with hardwired DRIVE and SR controls.
    - For AUDIOCLK: DRIVE = 0 and SR = 1
- HSIO I/Os with drive control only for the following lines:
  - SDMMC0: PA0, PA1, PA3, PA4, PA5, PA6, PA7, PA8, PA9, PA10, PA11
  - SDMMC1: PB29, PB30, PB31, PC0, PC1, PC2
  - SDMMC2: PD3, PD4, PD5, PD6, PD7, PD8
  - QSPI0: PB9, PB10, PB11, PB12, PB14, PB15, PB16, PB17, PB18, PB19, PB20

- PIOBU I/Os with no drive and no slewrate control

Unless otherwise specified:

- The following specifications apply to the GPIO, PIOBU and HSIO types.
- V_{DD} refers to the voltage of the associated power rail of the I/O line, as defined in the table Pin Description; for example, for PA12, V_{DD} refers to the voltage applied on VDDIOP0.

For SDMMC lines in UHS I mode in the 1.8V range, the drive is controlled in the SDMMC user interface. In this case, drive configurations in PIO registers have no effect.

For any other SDMMC configuration, the GPIO drive configuration is applicable.

DRIVE and SLEWRATE are set in the PIO_CFGR register corresponding to the GPIO, with the DRVSTR and SR bits, respectively.

74.5.1 I/O DC Characteristics

Table 74-15. Input DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Low level input voltage(1)	–	–	0.3 x V_{DD}	V
V_{IH}	High level input voltage(1)	–	0.7 x V_{DD}	–	V
I_{IH}	Input-high input leakage current(1)	Pull-down resistor disabled V_{IN} = V_{DD} = 3.6V	-150	150	nA
I_{IL}	Input-low input leakage current(1)	Pull-up resistor disabled V_{IN} = 0V, V_{DD} = 3.6V	-150	150	nA
R_{PULL}	Programmable pull-up or pull-down resistor	Digital Input mode	60	140	kΩ
C_{IN}	Input capacitance(1)	–	–	4	pF
V_{hys}	Input hysteresis(1)	–	150	–	mV

Note:

1. Simulation data

Table 74-16. Output DC Characteristics (1.7V < V_{DD} < 1.9V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
I_{OL} or I_{OH}	Low level or high level output current(1)	Any	I_{OL}: V_{OL} = 0.25 x VDD	–	–	–
			I_{OH}: V_{OH} = 0.75 x VDD	–	–	–
		GPIO	Drive = 0, Slewrate = 0	–	3	mA
			Drive = 1, Slewrate = 0	–	4	mA
			Drive = 0, Slewrate = 1	–	1	mA
			Drive = 1, Slewrate = 1	–	2	mA
		HSIO	–	–	1.5	mA
		PIOBU	–	–	2	mA

Note:

1. Simulation data

Table 74-17. Output DC Characteristics (3.0V < V_{DD} < 3.6V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
I_{OL} or I_{OH}	Low level/high level output current(1)	Any	I_{OL}: V_{OL} = 0.2 x V_{DD}	–	–	–
			I_{OH}: V_{OH} = 0.8 x V_{DD}	–	–	–
		GPIO	Drive = 0, Slewrate = 0	–	9	mA
			Drive = 1, Slewrate = 0	–	10	mA
			Drive = 0, Slewrate = 1	–	3	mA
			Drive = 1, Slewrate = 1	–	6	mA
		HSIO	–	–	2	mA
		PIOBU	–	–	5	mA

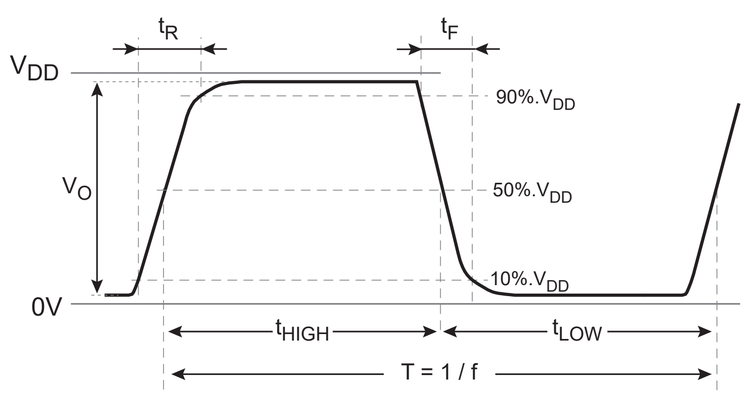
Note:
1. Simulation data

74.5.2 I/O AC Characteristics

74.5.2.1 Output Driver AC Characteristics

The timing definitions necessary to specify the maximum operating frequency of an output driver are provided in the following figure.

Figure 74-8. Timing Definitions of a Digital Output Signal



t_{SIG}: Period of the digital output signal

f = 1 / t_{SIG}: Frequency of the digital output signal

t_{HIGH}: Time during which the output waveform is greater than V_{DD} / 2

t_{LOW} = t_{SIG} - t_{HIGH}: Time during which the output waveform is less than V_{DD} / 2

d = t_{HIGH} / t_{SIG}: Output waveform duty cycle

V_O: Output waveform amplitude

In tables Output Driver AC Characteristics (1.7V < VDD < 1.9V, CL = 10 pF) and Output Driver AC Characteristics (3.0V < VDD < 3.6V, CL = 10 pF), the maximum operating frequency f_{MAX} ensures that the driver's output waveform fulfills the following characteristics:

- t_R < 0.75 / f_{MAX} and t_F < 0.75 / f_{MAX}
- d: the duty cycle of the output waveform is between 45% and 55%
- V_O: the output waveform amplitude is greater than 95% VDD

The f_{MAX} parameter indicates the speed limit of an output driver across various operating conditions: supply voltage range, load capacitance, drive strength programming. The effective maximum output frequency of a specific output line may be limited by the peripheral that drives this line.

Tables Output Driver AC Characteristics (1.7V < VDD < 1.9V, CL = 10 pF) and Output Driver AC Characteristics (3.0V < VDD < 3.6V, CL = 10 pF) provide the AC output characteristics of the output drivers in the following conditions:

- Output load: 10 pF capacitor to ground
- Two VDD ranges:
  - 1.7V < VDD < 1.9V and
  - 3.0V < VDD < 3.6V
- Two Drive settings: 0 and 1, and
- Two Slewrate settings for the GPIO type: 0 and 1.

For the GPIO drivers, the following table provides the recommended drive and slewrate settings depending on the output switching frequency and the two commonly used VDD ranges (1.8V and 3.3V). Other settings are possible but they may lead to excessively fast rise and fall times (t_R, t_F), with a potentially negative impact on the electromagnetic emissions of the application.

Table 74-18. Recommended GPIO Drive and Slewrate Settings vs IO Use Case

VDD Range	Low Speed $f_{GPIO} \leq$ 50 MHz(1)	High Speed 50 MHz $\leq f_{GPIO} \leq$ 170 MHz(1)
1.7V - 1.9V	Drive = 1, Slewrate = 1	Drive = 1, Slewrate = 0
3.0V - 3.6V	Drive = 0, Slewrate = 1	Drive = 0, Slewrate = 0

Note:

1. This is an indicative value. See the tables Output Driver AC Characteristics (1.7V < VDD < 1.9V, CL = 10 pF) and Output Driver AC Characteristics (3.0V < VDD < 3.6V, CL = 10 pF) for accurate maximum frequency specifications.

For the HSIO drivers (used in SDMMCx and QSPI0 peripherals), the following table provides their nominal output impedance with respect to the Drive settings. These drivers do not have a Slewrate setting but are rather calibrated against an external 1% resistor mounted on the SDMMCx_CAL or QSPI0_CAL pins. Depending on the target signal frequency and the external load, it is possible adjust their target output impedance.

Table 74-19. HSIO Output Impedance vs Drive Settings

VDD Range	Drive = 00 (Type D)	Drive = 01 (Type A)	Drive = 10 (Type C)	Drive = 11 (Type B)	Unit
1.7V – 1.9V	100	33	66	50	Ω
3.0V – 3.6V	84	28	56	42	Ω

Note: For SDMMC lines in UHS I mode in 1.8V, the drive is controlled in the SDMMC user interface. In this case, drive configurations in PIO registers have no effect.

Table 74-20. Output Driver AC Characteristics (1.7V < V_{DD} < 1.9V, C_L = 10 pF)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
t_R or t_F	Rise or fall time[(1)(2)]	GPIO	Drive = 0, Slewrate = 0	1.5	5.1	ns
			Drive = 1, Slewrate = 0	1.4	4.8	ns
			Drive = 0, Slewrate = 1	4.1	12.3	ns
			Drive = 1, Slewrate = 1	2.1	7.0	ns
		HSIO	Drive = 11 (Type B)	1.6	1.8	ns
			Drive = 01 (Type A)	1.1	1.25	ns
			Drive = 10 (Type C)	2.2	2.3	ns
			Drive = 00 (Type D)	3.3	3.4	ns
		PIOBU	–	1.0	4.0	ns
f_{MAX}	Maximum frequency[(2)(3)]	GPIO	Drive = 0, Slewrate = 0	–	95	MHz
			Drive = 1, Slewrate = 0	–	135	MHz
			Drive = 0, Slewrate = 1	–	25	MHz
			Drive = 1, Slewrate = 1	–	50	MHz
		HSIO	Drive = 11 (Type B)	–	125	MHz
			Drive = 01 (Type A)	–	150	MHz
			Drive = 10 (Type C)	–	100	MHz
			Drive = 00 (Type D)	–	70	MHz
		PIOBU	–	–	25	MHz

Notes:
1. Measured between V_O = 10% V_{DD} and V_O= 90% V_{DD}
2. Simulation data
3. f_{MAX} may be limited by the peripheral that drives the I/O line.

Table 74-21. Output Driver AC Characteristics (3.0V < V_{DD} < 3.6V, C_L = 10 pF)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
t_R or t_F	Rise or fall time[(1)(2)]	GPIO	Drive = 0, Slewrate = 0	1.5	2.9	ns
			Drive = 1, Slewrate = 0	1.4	2.7	ns
			Drive = 0, Slewrate = 1	4.0	7.0	ns
			Drive = 1, Slewrate = 1	2.2	4.0	ns
		HSIO	Drive = 11 (Type B)	1.3	1.5	ns
			Drive = 01 (Type A)	0.9	1.0	ns
			Drive = 10 (Type C)	1.7	1.9	ns
			Drive = 00 (Type D)	2.6	2.9	ns
		PIOBU	–	1.0	3.0	ns

...........continued

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
f_{MAX}	Maximum frequency[(2)(3)]	GPIO	Drive = 0, Slewrate = 0	–	150	MHz
			Drive = 1, Slewrate = 0	–	170	MHz
			Drive = 0, Slewrate = 1	–	50	MHz
			Drive = 1, Slewrate = 1	–	75	MHz
		HSIO	Drive = 11 (Type B)	–	125	MHz
			Drive = 01 (Type A)	–	150	MHz
			Drive = 10 (Type C)	–	100	MHz
			Drive = 00 (Type D)	–	70	MHz
		PIOBU	–	–	40	MHz

Notes:
1. Measured between V_O = 10% V_{DD} and V_O= 90% V_{DD}
2. Simulation data
3. f_{MAX} may be limited by the peripheral that drives the I/O line.

74.5.2.2 Input AC Characteristics

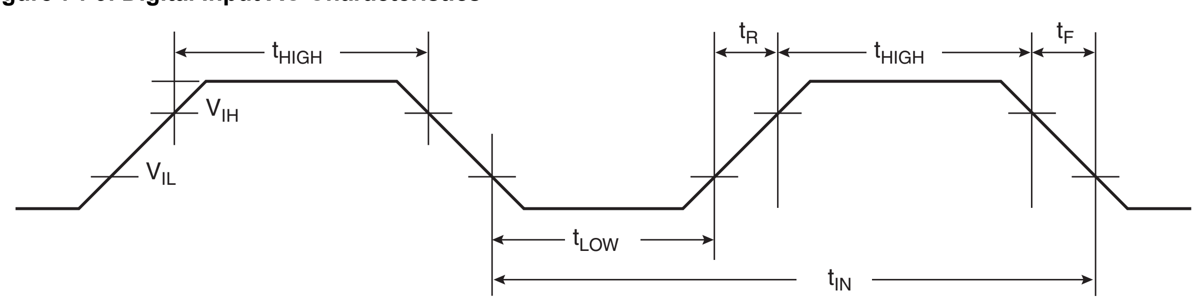
The following table provides the input characteristics of the I/O lines when configured as a digital input. In particular, these values apply when the XIN input is used as a clock input of the device (main crystal oscillator set in Bypass mode). They do not apply for the XIN32 input which is designed for slow signals with frequencies up to 50 kHz. Parameters V_{IL} and V_{IH} defined in the table Input DC Characteristics apply.

Table 74-22. Input AC Characteristics

Symbol	Parameter	Min	Max	Units
f_{IN}	Input frequency[(1)]	–	50	MHz
t_{IN}	Input period	20	–	ns
t_{HIGH}	Time at high level	8	–	ns
t_{LOW}	Time at low level	8	–	ns
t_R	Rise time	–	2.2	ns
t_F	Fall time	–	2.2	ns

Note:
1. The maximum input frequency may be limited by the peripheral receiving this signal.

Figure 74-9. Digital Input AC Characteristics



74.5.3 DDR I/O Calibration and DDR Voltage Reference

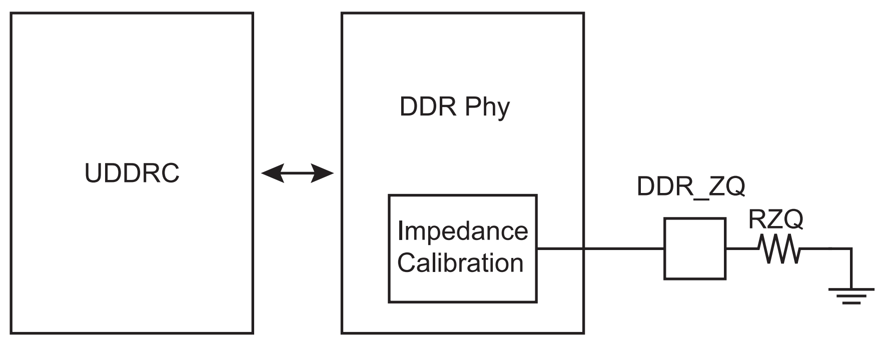
74.5.3.1 DDR I/O Calibration

The DDR2/DDR3/LPDDR2/LPDDR3/DDR3L I/Os embed an automatic impedance matching control to avoid overshoots and reach the best performance levels depending on the bus load and external memories. A serial termination connection scheme, where the driver has an output impedance matched to the characteristic impedance of the line, is used to improve signal quality and reduce EMI.

One specific analog input, DDR_ZQ, is used to calibrate all DDR IOs with the external resistor RZQ = 240 Ω ±1%.

The UDDRC supports the ZQ calibration procedure used to calibrate the device DDR I/O drive strength and the commands to setup the external DDR device drive strength (refer to the section Universal DDR Memory Controller (UDDRC). The calibration cell supports all the memory types listed above.

Figure 74-10. DDR Calibration Cell



74.5.3.2 DDR Voltage Reference

The DDR_VREF input must be driven at all times by a voltage source equal to VDDIODDR/2. See below the recommended schematics to drive DDR_VREF.

Figure 74-11. DDR_VREF Recommended Schematic with DDR2/3(L)-SDRAM

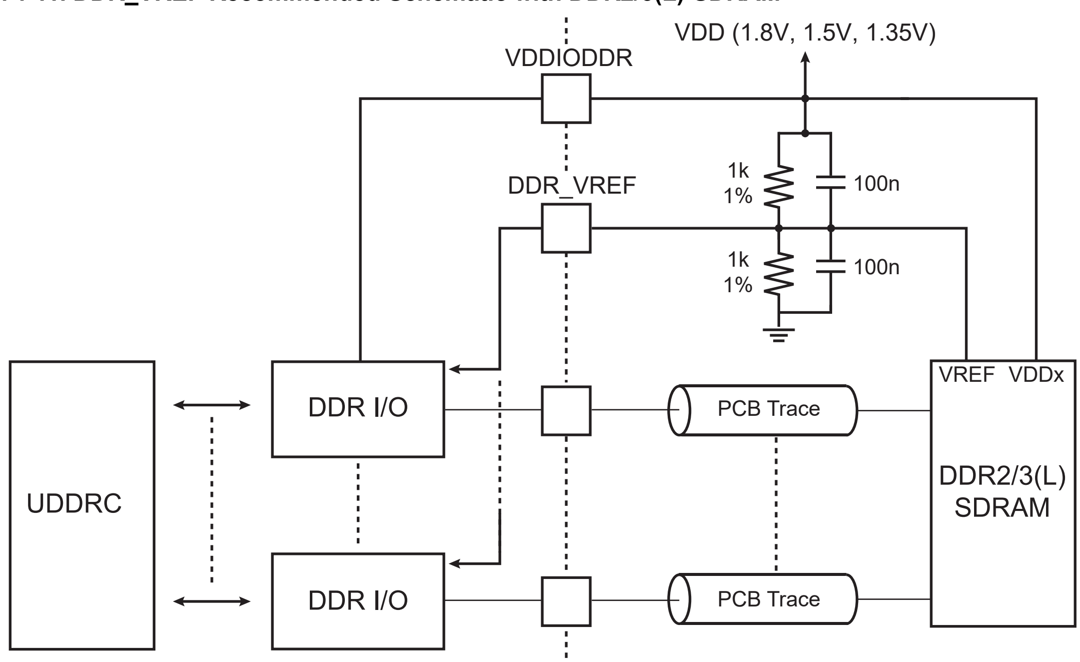
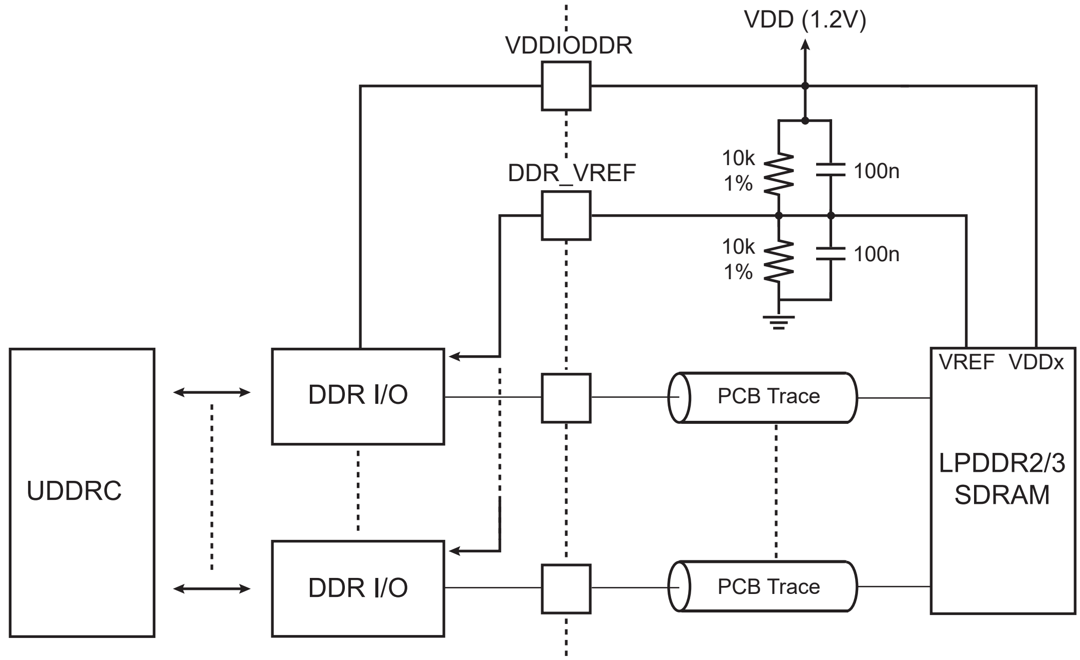


Figure 74-12. DDR_VREF Recommended Schematic with LPDDR2/3-SDRAM

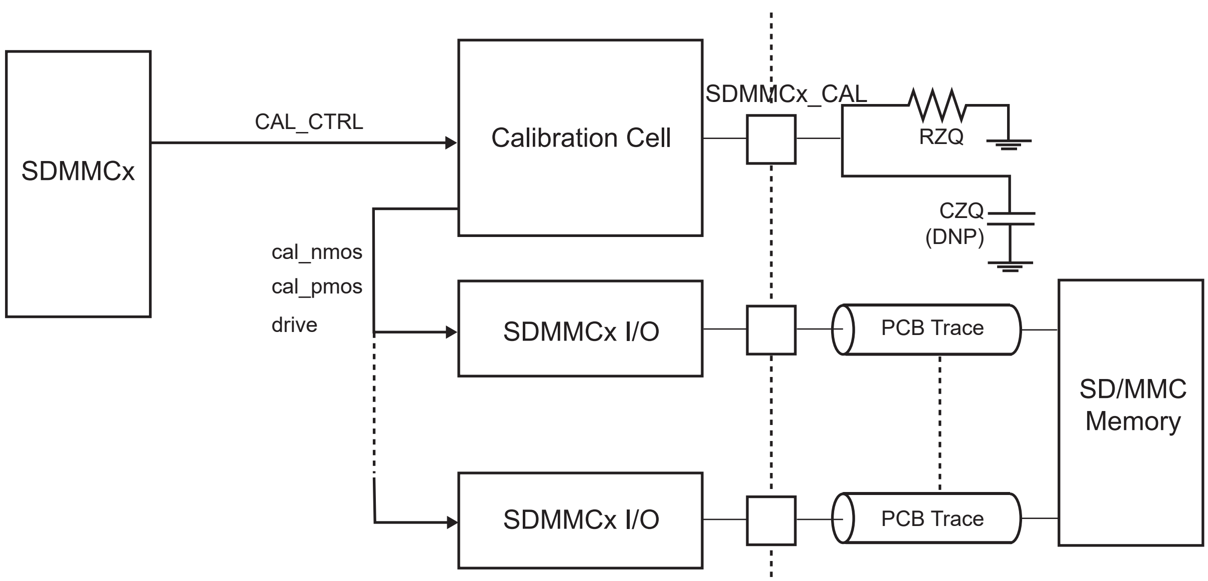


74.5.4 SDMMC and QSPI I/O Calibration

74.5.4.1 SDMMC I/O Calibration

The device embeds I/O calibration cells for SDMMC0, SDMMC1 and SDMMC2. The purpose of this block is to provide to e.MMC/SD I/Os an output impedance reference to limit the impact of process, voltage and temperature on the drivers output impedance. The impedance control is required at high frequency in order to improve signal quality.

Figure 74-13. SDMMCx I/O Calibration Cell



The calibration cell provides input pin SDMMCx_CAL loaded with a 20 KΩ RZQ resistor for 1.8V memories and a 16.9 KΩ resistor for 3.3V memories. In the above figure, CZQ is not mounted.

- According to the e.MMC specification, the output impedance calibration is required for HS200 and HS400 modes (1.8V) whereas it is not for other modes (3.3V).
- In addition, according to the SD specification, the output impedance calibration is required for 1.8V signaling in SD UHS-I whereas it is not for 3.3V signaling. When the interface needs to operate at both 1.8V and 3.3V, RZQ has the 1.8V value (20 KΩ) .

The following table shows the values to program in the SDMMCx_CALCR.CLKDIV field with respect to MCK1 clock frequency.

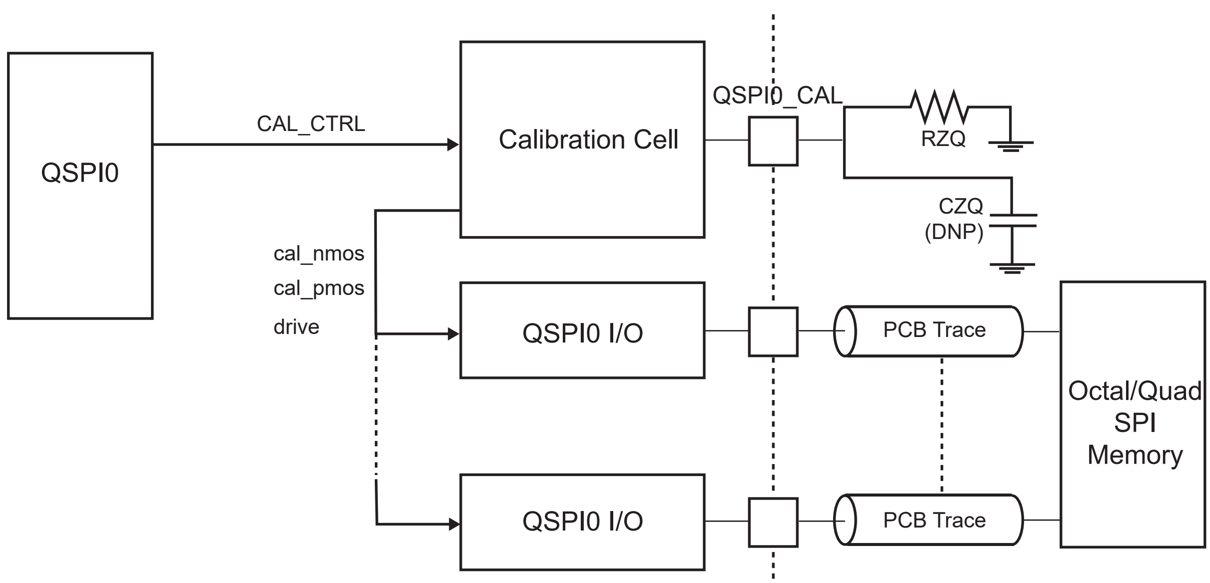
Table 74-23. SDMMCx_CALCR.CLKDIV vs MCK1 Clock Frequency

CLKDIV	Divider	MCK1 (MHz)
0	2	25
1	4	50
2	6	75
3	8	100
4	10	125
5	12	150
6	14	175
7	16	200

74.5.4.2 QSPI0 I/O Calibration

The device embeds a QSPI0 I/O calibration cell. The purpose of this block is to provide to octal/quad SPI I/Os an output impedance reference to limit the impact of process, voltage and temperature on the drivers output impedance. The impedance control is required at high frequency in order to improve signal quality.

Figure 74-14. QSPI0 I/O Calibration Cell



The calibration cell provides an input pin QSPI0_CAL loaded with a 20 KΩ RZQ resistor for 1.8V memories and a 16.9 KΩ resistor for 3.3V memories. In the above figure, CZQ is not mounted.

According to the QSPI specification, the output impedance calibration is mandatory for QSPI Hyperflash mode (1.8V), whereas it is not for other modes (3.3V).

The following table provides the values to program in the QSPI_PCALCFG.CLKDIV field with respect to MCK1 clock frequency.

Table 74-24. QSPI_PCALCFG.CLKDIV vs MCK1 Clock Frequency

CLKDIV	Divider	MCK1 (MHz)
0	2	25
1	4	50
2	6	75
3	8	100
4	10	125
5	12	150
6	14	175
7	16	200

74.5.5 QSPI Characteristics

Figure 74-15. QSPI Host Mode 0 in Single Data Rate

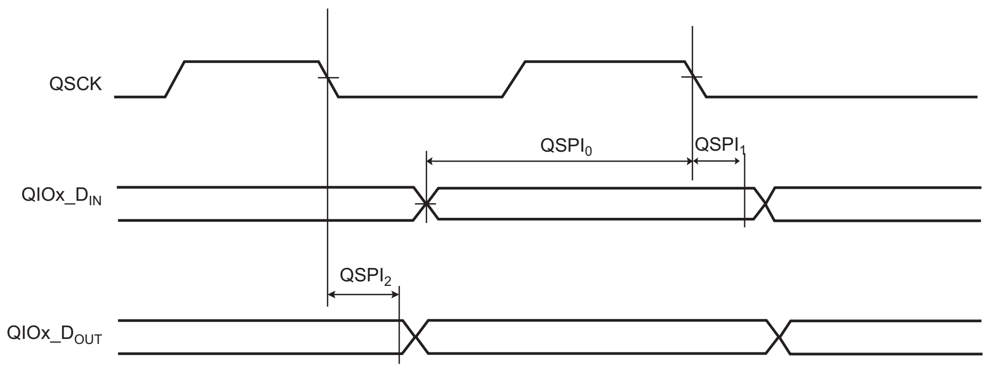
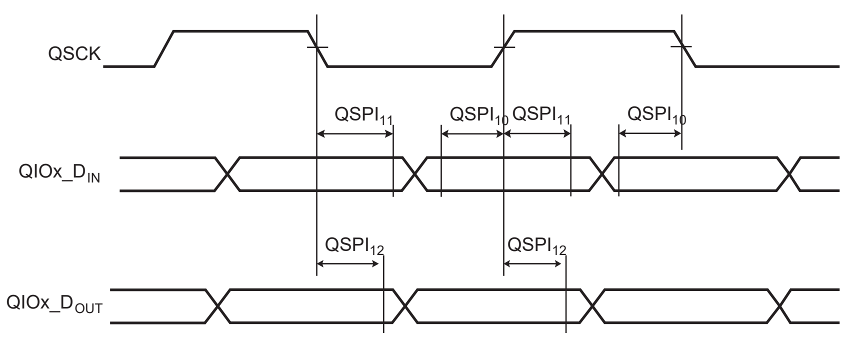


Figure 74-16. QSPI Host Mode in Double Data Rate Mode



74.5.5.1 Maximum QSPI Frequency

The QSPI GCLK frequency must be set to the target frequency f_{QSCK}. The following formulas provide the achievable QSPI frequency in Host Read and Host Write modes. When the result of these formulas exceeds the f_{QSCK_MAX} parameter provided in the tables in QSPI Timings, f_{QSCK_MAX} applies.

- **Host Write in Single Data Rate Mode**

$$f_{\mathrm{QSCK}} = \frac{1}{(\mathrm{QSPI}_2 + t_{\mathrm{SU\_CLIENT}})}$$

Where t_{SU_CLIENT} is the input setup time of the client device.

- **Host Read in Single Data Rate Mode**

$$f_{QSCK} = \frac{1}{QSPI_0 + t_{VALID}}$$

Where t_{VALID} is the client time response to output data after detecting a QSCK edge.

  – For a QSPI client device with t_{VALID} (or t_V) = 12 ns, with $QSPI_0$ = 1.0 ns, f_{QSCK} max = 77 MHz.
  – For a QSPI Flash memory device with t_{VALID} (or t_V) = 6 ns, and $QSPI_0$ = 1.0 ns, the formula returns a value of 143 MHz, therefore f_{QSCK} = 133 MHz applies in this case.

- **Host Write in Dual Data Rate Mode**

$$f_{QSCK} = \frac{0.5}{(QSPI_{12} + t_{SU\_CLIENT})}$$

Where t_{SU_CLIENT} is the input setup time of the client device.

- **Host Read in Dual Data Rate Mode**

$$f_{QSCK} = \frac{1}{2\text{x}(\mathrm{QSPI}_{10} + t_{VALID})}$$

Where t_{VALID} is the client time response to output data after detecting a QSCK edge.

  – For a QSPI Flash memory device with t_{VALID} (or t_V) = 6 ns, and $QSPI_{10}$ = 1.0 ns, the formula returns a value of 71 MHz.

74.5.5.2 QSPI Timings

For the QSPI0 instance, the timings shown in the tables below are provided in the following domains:

- 1.8V domain: VDDQSPI0 from 1.7V to 1.9V, maximum external capacitor: 10 pF, drive: 01 (Type A)
- 3.3V domain: VDDQSPI0 from 3.0V to 3.6V, maximum external capacitor: 10 pF, drive: 01 (Type A)

For the QSPI1 instance, the timings shown in the table below are provided in the following domains:

- 1.8V domain: VDDQSPI1 from 1.7V to 1.9V, maximum external capacitor: 10 pF, DRV: 1, SR: 1
- 3.3V domain: VDDQSPI1 from 3.0V to 3.6V, maximum external capacitor: 10 pF, DRV: 0, SR: 1

Table 74-25. QSPI Timings in Single Data Rate Mode (STR)

Symbol	Parameter	Conditions	Min	Max	Unit
f_{QSCK}	QSCK maximum frequency	–	–	133	MHz
$QSPI_0$	QIOx data in to QSCK falling edge (input setup time)	–	1	–	ns
$QSPI_1$	QIOx data in to QSCK falling edge (input hold time)	–	1	–	ns
$QSPI_2$	QSCK falling edge to QIOx data out valid	–	0	2	ns

Table 74-26. QSPI Timings in Double Data Rate Mode (DTR)

Symbol	Parameter	Conditions	Min	Max	Unit
f_{QSCK}	QSPI maximum frequency	–	–	75	MHz
$QSPI_{10}$	QIOx data in to QSCK edge (rising or falling, input setup time)	–	1	–	ns
$QSPI_{11}$	QIOx data in to QSCK edge (rising or falling, input hold time)	–	1	–	ns
$QSPI_{12}$	QSCK edge (rising or falling) to QIOx data out valid	–	$t_{QSCK}/4$ -2	$t_{QSCK}/4$ +2	ns

74.5.6 FLEXCOM SPI Characteristics

In the figures FLEXCOM in SPI Host Mode 1 and 2 and FLEXCOM in SPI Host Mode 0 and 3 below, the MOSI line shifting edge is represented with a hold time equal to 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown in the figure FLEXCOM in SPI Mode: MISO Capture in Host Mode, the device sampling point extends the propagation delay (t_p) for client and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI client working in Mode 0 can be safely driven if the SPI host is configured in Mode 0.

Figure 74-17. FLEXCOM in SPI Mode: MISO Capture in Host Mode

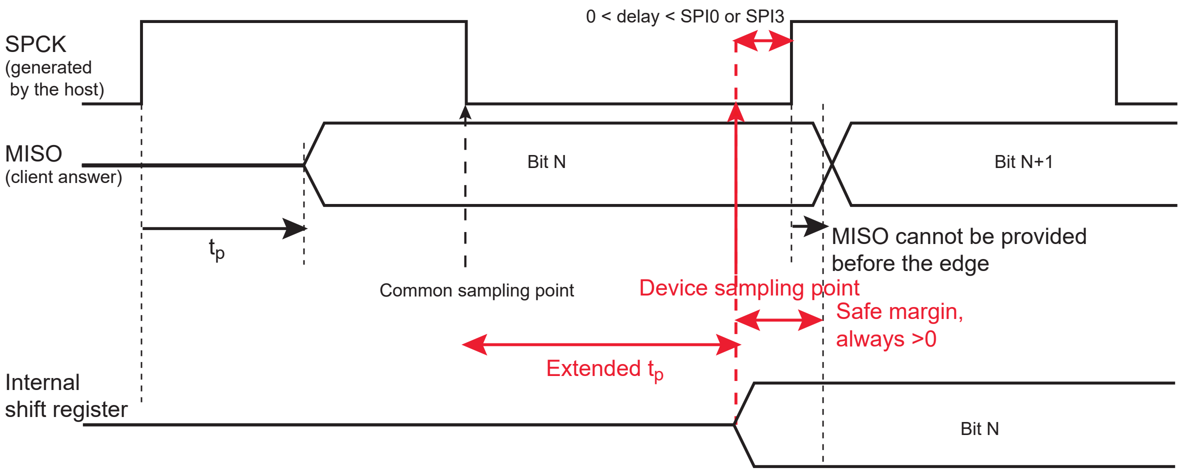


Figure 74-18. FLEXCOM in SPI Host Mode 1 and 2

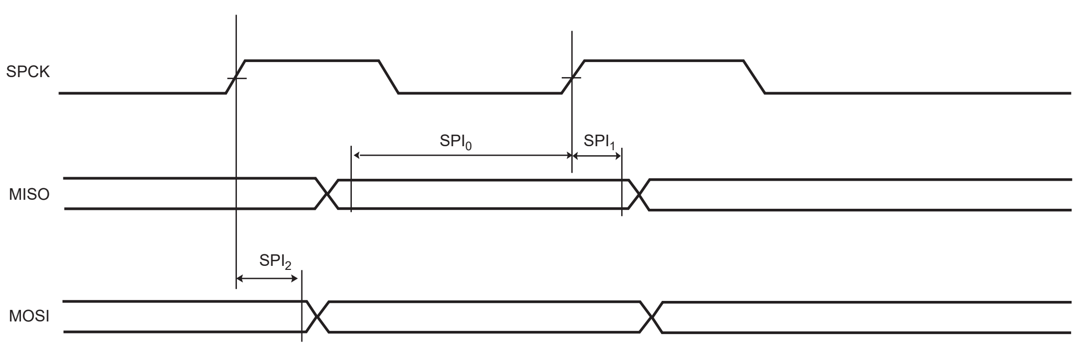


Figure 74-19. FLEXCOM in SPI Host Mode 0 and 3

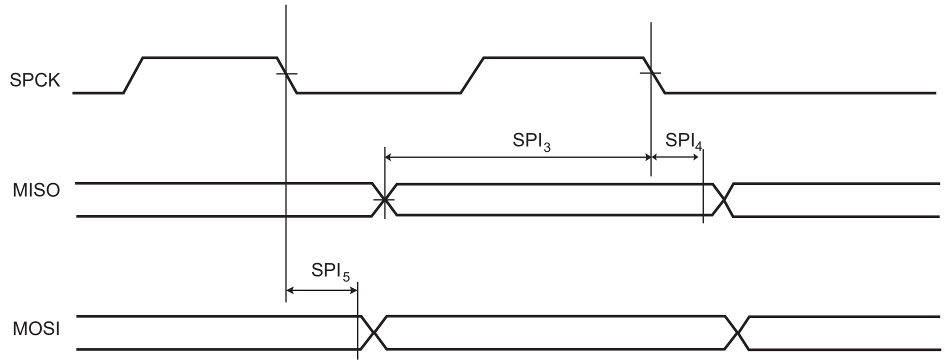


Figure 74-20. FLEXCOM in SPI Client Mode 0 and 3

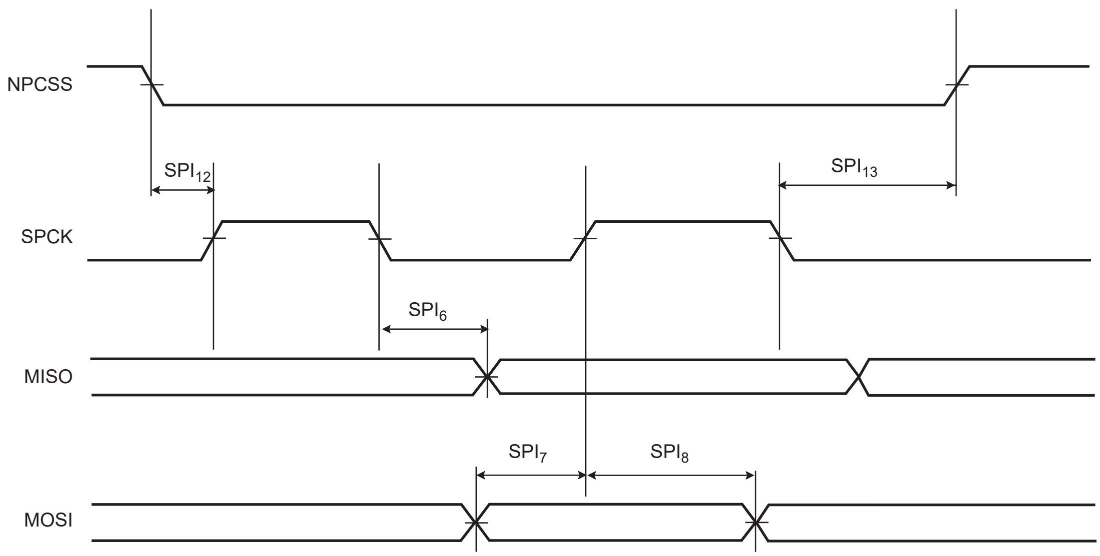


Figure 74-21. FLEXCOM in SPI Client Mode 1 and 2

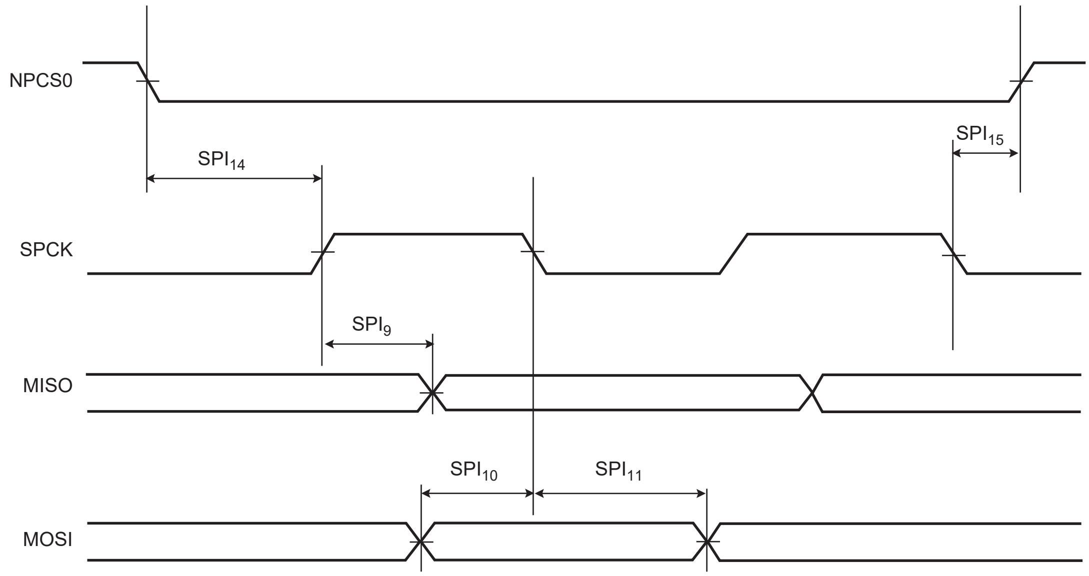
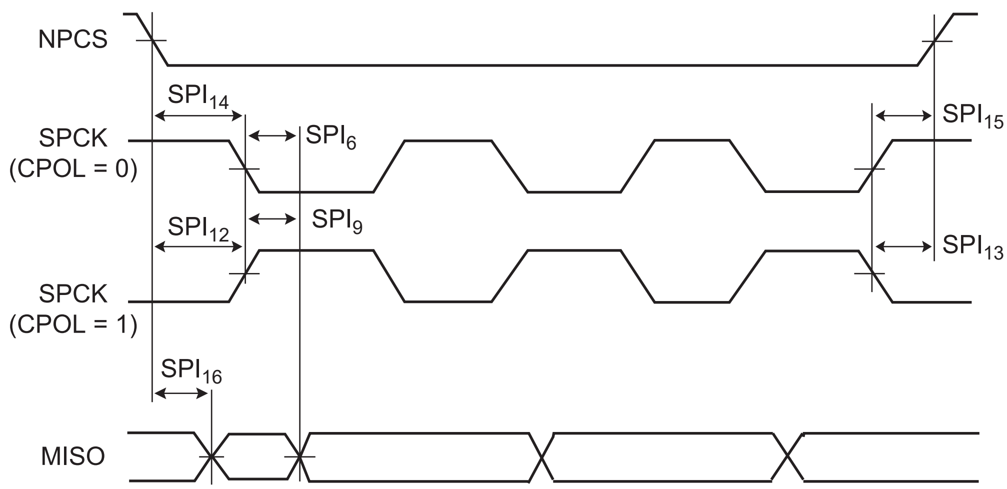


Figure 74-22. FLEXCOM in SPI Client - NPCS Timings



74.5.6.1 Maximum FLEXCOM SPI Frequency

The following formulas provide the achievable SPI frequency in Host Read and Write modes and in Client Read and Write modes. When the result of these formulas exceeds the f_{SPCK_MAX} parameter provided in the FLEXCOM SPI Timings table, f_{SPCK_MAX} applies.

74.5.6.1.1 Host Write Mode

$$f_{\mathrm{SPCK}} = \frac{1}{(\mathrm{SPI2\ or\ SPI5} + t_{\mathrm{SU\_CLIENT}})}$$

Where t_{SU_CLIENT} is the input setup time of the client device.

74.5.6.1.2 Host Read Mode

$$f_{\mathrm{SPCK}} = \frac{1}{\mathrm{SPI}_0(\mathrm{or\ SPI}_3) + t_{\mathrm{VALID}}}$$

Where t_{VALID} is the client time response to output data after detecting an SPCK edge.

For a nonvolatile memory with t_{VALID} (or t_v) = 5 ns, using SPI_3, f_{SPCK} = 40 MHz.

74.5.6.1.3 Client Read Mode

$$f_{\mathrm{SPCK}} = \frac{0.5}{(\mathrm{SPI}_7\ \ \mathrm{or}\ \ \mathrm{SPI}_{10} + t_{\mathrm{VALID\_HOST}})}$$

Where t_{VALID_HOST} is the host's delay between the SPCK edge and the output data valid.

74.5.6.1.4 Client Write Mode

$$f_{\mathrm{SPCK}} = \frac{1}{2x(SPI_{6\mathrm{max}}(\mathrm{or\ SPI}_{9\mathrm{max}}) + t_{\mathrm{setup}})}$$

Where t_{setup} is the setup time from the host before sampling data.

74.5.6.2 FLEXCOM SPI Timings

The timings shown in the table below are provided in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV = 0, SR = 1

Table 74-27. FLEXCOM SPI Timings

Symbol	Parameter	Min	Max	Unit
Host Mode				
f_{SPCK}	Frequency in Host mode	–	40	MHz
SPI_0	MISO setup time before SPCK rises	20	–	ns
SPI_1	MISO hold time after SPCK rises	0	–	ns
SPI_2	SPCK rising to MOSI delay	0	10	ns
SPI_3	MISO setup time before SPCK falls	20	–	ns
SPI_4	MISO hold time after SPCK falls	0	–	ns
SPI_5	SPCK falling to MOSI delay	0	10	ns
Client Mode				
f_{SPCK}	Frequency in Client mode	–	20	MHz
SPI_6	SPCK falling to MISO delay	4	20	ns
SPI_7	MOSI setup time before SPCK rises	3	–	ns
SPI_8	MOSI hold time after SPCK rises	3	–	ns
SPI_9	SPCK rising to MISO delay	4	20	ns
SPI_{10}	MOSI setup time before SPCK falls (client)	3	–	ns
SPI_{11}	MOSI hold time after SPCK falls (client)	3	–	ns
SPI_{12}	NPCS setup to SPCK rising (client)	6	–	ns
SPI_{13}	NPCS hold after SPCK falling (client)	1	–	ns
SPI_{14}	NPCS setup to SPCK falling (client)	6	–	ns
SPI_{15}	NPCS hold after SPCK falling (client)	1	–	ns

...........continued				
Symbol	**Parameter**	**Min**	**Max**	**Unit**
SPI_{16}	NPCS falling to MISO valid (client)	20	–	ns

74.5.7 FLEXCOM USART Characteristics

Table 74-28. FLEXCOM USART Timings

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{SCK}	SCK frequency	Synchronous (SYNC=1) host or client	–	–	25	MHz
f_S	UART sampling clock frequency	Asynchronous (SYNC=0)[1]	–	–	25	MHz

Note:

1. The corresponding bit rate is $R = f_S / 8 = 3.1$ Mbits/s with oversampling FLEX_US_MR.OVER=1 and $R = f_S / 16 = 1.5$ Mbits/s with oversampling FLEX_US_MR.OVER=0. Refer to the Baud Rate Generator figure in the section Flexible Serial Communication Controller (FLEXCOM).

74.5.8 SDMMC Characteristics

The standards below applies in the following domains:

- 1.8V domain:
  - VDDSDMMC: from 1.7V to 1.9V
  - Maximum external capacitor: 10 pF
  - Drive: 01 (Type A)
- 3.3V domain:
  - VDDSDMMC: from 3.0V to 3.6V
  - Maximum external capacitor: 10 pF
  - Drive: 01 (Type A)

The Secure Digital Multimedia Card Controller (SDMMC0) complies with the following specifications:

- SD Host Controller Standard Specification Version 3.00
- SD Memory Card Specification Version 3.00
- SDIO Specification Version 3.00
  - Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
  - Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)
  - Support for SD/SDIO UHS-I SDR12 (Maximum SDCLK Frequency = 25 MHz)
  - Support for SD/SDIO UHS-I SDR25 (Maximum SDCLK Frequency = 50 MHz)
  - Support for SD/SDIO UHS-I SDR50 (Maximum SDCLK Frequency = 100 MHz)
  - Support for SD/SDIO UHS-I SDR104 (Maximum SDCLK Frequency = 200 MHz)
  - Support for SD/SDIO UHS-I DDR50 (Maximum SDCLK Frequency = 50 MHz)
- MultiMedia Card Specification Version V5.1
  - Support for MMC/e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
  - Support for MMC/e.MMC High Speed (Maximum SDCLK Frequency = 52 MHz)
  - Support for e.MMC High Speed DDR (Maximum SDCLK Frequency = 52 MHz)
  - Support for e.MMC HS200 (Maximum SDCLK Frequency = 200 MHz SDR)
  - Support for e.MMC HS400 (Maximum SDCLK Frequency = 200 MHz DDR)

The Secure Digital Multimedia Card Controllers (SDMMC1 and SDMMC2) comply with the following specifications:

- SD Host Controller Standard Specification Version 3.00
- SD Memory Card Specification Version 3.00
- SDIO Specification Version 3.00
  - Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
  - Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)

- Support for SD/SDIO UHS-I SDR12 (Maximum SDCLK Frequency = 25 MHz)
- Support for SD/SDIO UHS-I SDR25 (Maximum SDCLK Frequency = 50 MHz)
- Support for SD/SDIO UHS-I SDR50 (Maximum SDCLK Frequency = 100 MHz)
- Support for SD/SDIO UHS-I SDR104 (Maximum SDCLK Frequency = 200 MHz)
- Support for SD/SDIO UHS-I DDR50 (Maximum SDCLK Frequency = 50 MHz)

- MultiMedia Card Specification Version V4.51
  - Support for MMC/e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
  - Support for MMC/e.MMC High Speed (Maximum SDCLK Frequency = 52 MHz)
  - Support for e.MMC High Speed DDR (Maximum SDCLK Frequency = 52 MHz)
  - Support for e.MMC HS200 (Maximum SDCLK Frequency = 200 MHz SDR)

74.5.9 UDDRC and DDR3PHY Characteristics

The UDDRC and DDR3PHY are compliant with the following JEDEC standards:

- DDR2-SDRAM: JESD79-2E with operating frequencies up to 533 MHz
- DDR3-SDRAM: JESD79-3C with operating frequencies up to 533 MHz
- DDR3L-SDRAM: JESD79-3-1A with operating frequencies up to 533 MHz
- LPDDR2-SDRAM: JESD209-2B with operating frequencies up to 533 MHz
- LPDDR3-SDRAM: JESD209-3B with operating frequencies up to 533 MHz

The physical interface (PCB layout) between the processor and its memory has a major impact on signal integrity. Microchip provides IBIS models of the SAMA7G5 device and strongly recommends to verify this processor-memory interface on a PCB simulation tool.

74.5.10 SMC Timings

Timings are provided in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV= 1, SR = 1
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV= 0, SR = 1

74.5.10.1 Read Timings

Table 74-29. SMC Read Signals - NRD-Controlled (READ_MODE = 1)

Symbol	Parameter	Min	Max	Unit
NO HOLD SETTINGS (NRD_HOLD = 0)				
SMC_1	Data setup before NRD high	17	–	ns
SMC_2	Data hold after NRD high	0	–	ns
HOLD SETTINGS (NRD_HOLD ≠ 0)				
SMC_3	Data setup before NRD high	13	–	ns
SMC_4	Data hold after NRD high	0	–	ns
HOLD or NO HOLD SETTINGS (NRD_HOLD ≠ 0, NRD_HOLD = 0)				
SMC_5	A0–A22 Valid before NRD High	(NRD_SETUP + NRD_PULSE) × t_{MCK1} - 7	–	ns
SMC_6	NCS low before NRD High	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{MCK1} - 3	–	ns
SMC_7	NRD pulse width	NRD_PULSE × t_{MCK1}	–	ns

Table 74-30. SMC Read Signals - NCS-Controlled (READ_MODE= 0)

Symbol	Parameter	Conditions	Min	Max	Unit
NO HOLD SETTINGS (NCS_RD_HOLD = 0)					

...........continued

Symbol	Parameter	Conditions	Min	Max	Unit
SMC_8	Data setup before NCS high	–	19	–	ns
SMC_9	Data hold after NCS high	–	0	–	ns
HOLD SETTINGS (NCS_RD_HOLD ≠ 0)					
SMC_{10}	Data setup before NCS high	–	17	–	ns
SMC_{11}	Data hold after NCS high	–	0	–	ns
HOLD or NO HOLD SETTINGS (NCS_RD_HOLD ≠ 0, NCS_RD_HOLD = 0)					
SMC_{12}	A0–A22 Valid before NCS High	–	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{MCK1} - 6	–	ns
SMC_{13}	NRD low before NCS High	–	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{MCK1}	–	ns
SMC_{14}	NCS pulse width	–	NCS_RD_PULSE × t_{MCK1}	–	ns

74.5.10.2 Write Timings

Table 74-31. SMC Write Signals - NWE-Controlled (WRITE_MODE = 1)

Symbol	Parameter	Min	Max	Unit
HOLD or NO HOLD SETTINGS (NWE_HOLD ≠ 0, NWE_HOLD = 0)				
SMC_{15}	Data out valid before NWE high	NWE_PULSE × t_{MCK1} - 9	–	ns
SMC_{16}	NWE pulse width	NWE_PULSE × t_{MCK1}	–	ns
SMC_{17}	A0–A22 valid before NWE low	NWE_SETUP × t_{MCK1} - 7	–	ns
SMC_{18}	NCS low before NWE high	(NWE_SETUP - NCS_RD_SETUP + NWE_PULSE) × t_{MCK1} - 4	–	ns
HOLD SETTINGS (NWE_HOLD ≠ 0)				
SMC_{19}	NWE high to data OUT, NBS0/A0 NBS1, A1 - A23 change	NWE_HOLD × t_{MCK1}	–	ns
SMC_{20}	NWE high to NCS inactive(1)	(NWE_HOLD - NCS_WR_HOLD) × t_{MCK1}	–	ns
NO HOLD SETTINGS (NWE_HOLD = 0)				
SMC_{21}	NWE high to data OUT, NBS0/A0 NBS1, A1 - A23, NCS change(1)	3	–	ns

Note:

1. Hold length = Total cycle duration - setup duration - pulse duration. “hold length” is for “NCS_WR_HOLD length” or “NWE_HOLD length”.

Table 74-32. SMC Write Signals - NCS-Controlled (WRITE_MODE = 0)

Symbol	Parameter	Min	Max	Unit
SMC_{22}	Data out valid before NCS high	NCS_WR_PULSE × t_{MCK1} - 6	–	ns
SMC_{23}	NCS pulse width	NCS_WR_PULSE × t_{MCK1}	–	ns
SMC_{24}	A0–A22 valid before NCS low	NCS_WR_SETUP × t_{MCK1} - 6	–	ns

...........continued				
Symbol	Parameter	Min	Max	Unit
SMC_{25}	NWE low before NCS high	(NCS_WR_SETUP - NWE_SETUP + NCS_PULSE) × t_{MCK1}	–	ns
SMC_{26}	NCS high to data OUT, A0 - A25 change	NCS_WR_HOLD × t_{MCK1} - 6	–	ns
SMC_{27}	NCS high to NWE inactive	(NCS_WR_HOLD - NWE_HOLD) × t_{MCK1} - 6	–	ns

Figure 74-23. SMC Timings - NCS-Controlled Read and Write

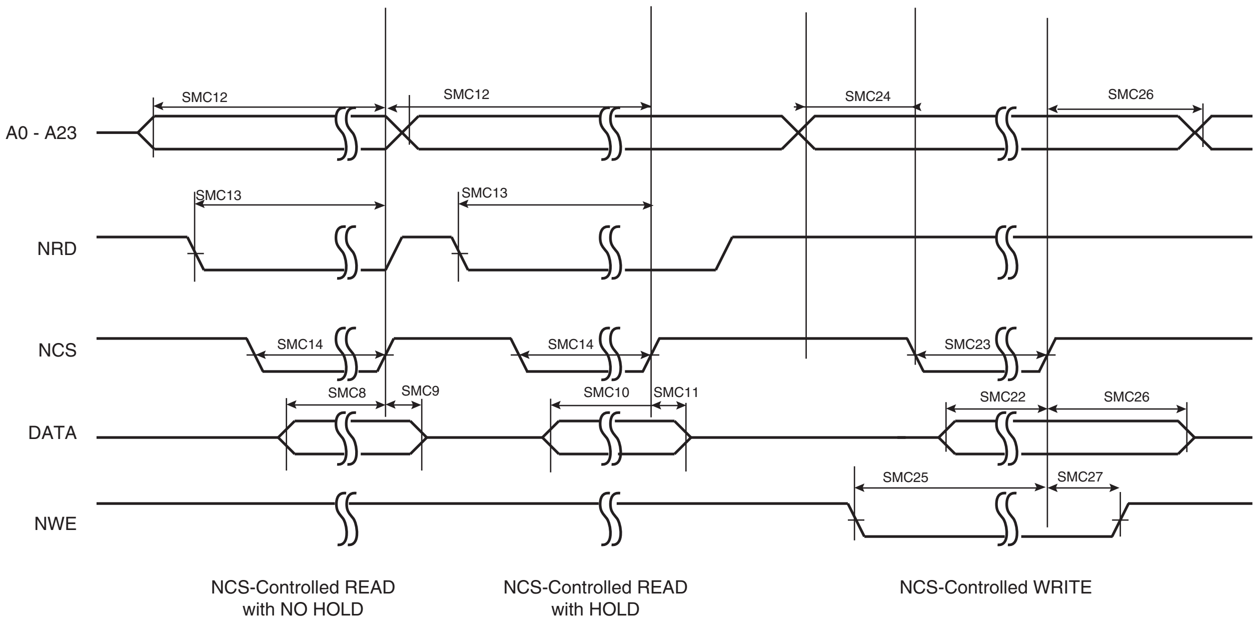
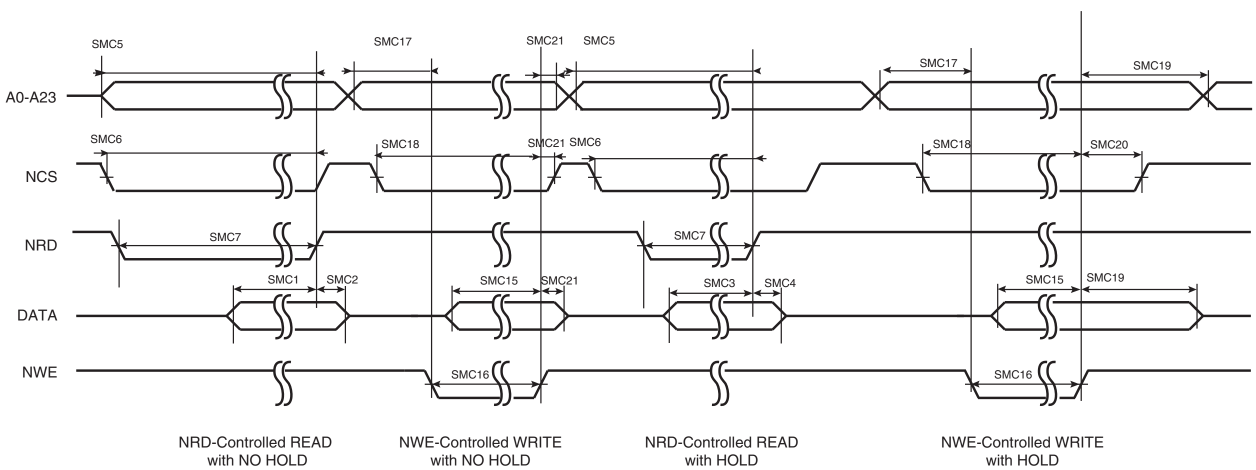


Figure 74-24. SMC Timings - NRD Controlled Read and NWE Controlled Write



74.5.11 SSC Timings

Timings are provided in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV= 1, SR = 1
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV= 0, SR = 1

Figure 74-25. SSC Transmitter, TK and TF in Output

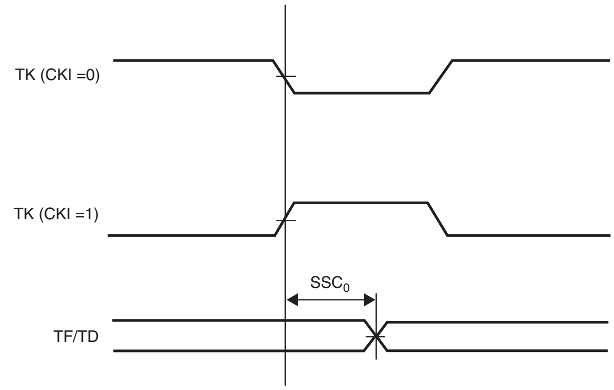


Figure 74-26. SSC Transmitter, TK in Input and TF in Output

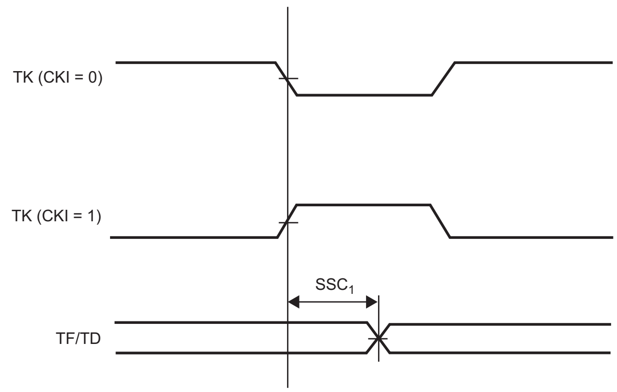


Figure 74-27. SSC Transmitter, TK in Output and TF in Input

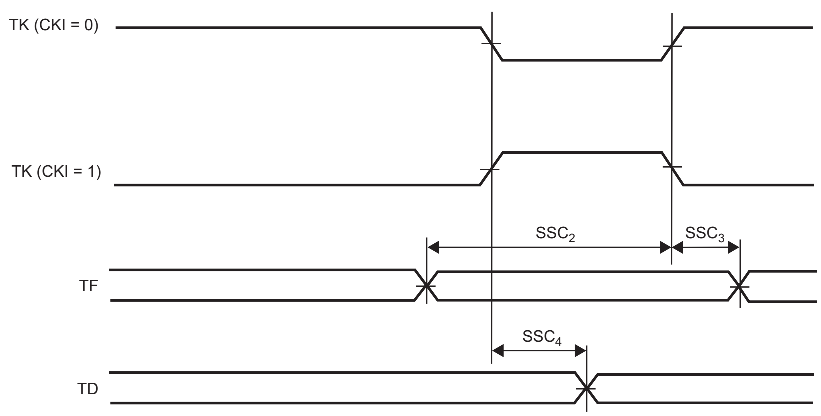


Figure 74-28. SSC Transmitter, TK and TF in Input

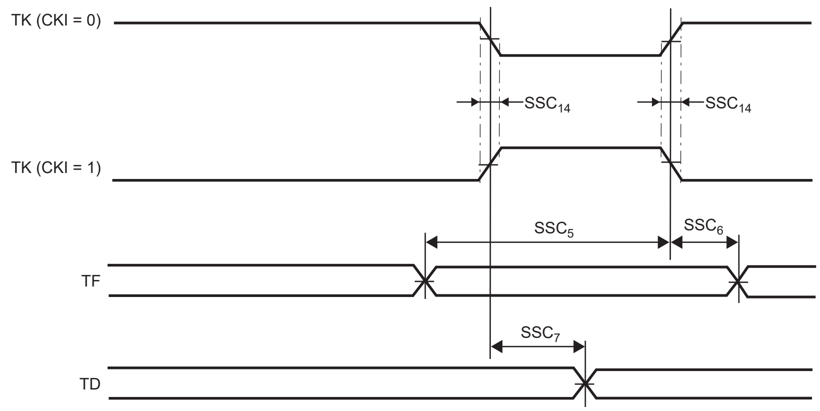


Figure 74-29. SSC Receiver RK and RF in Input

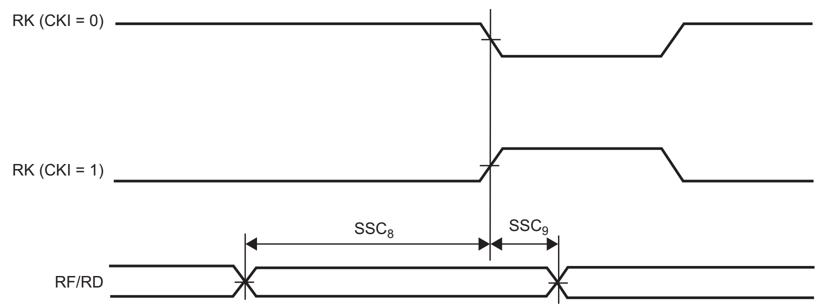


Figure 74-30. SSC Receiver, RK in Input and RF in Output

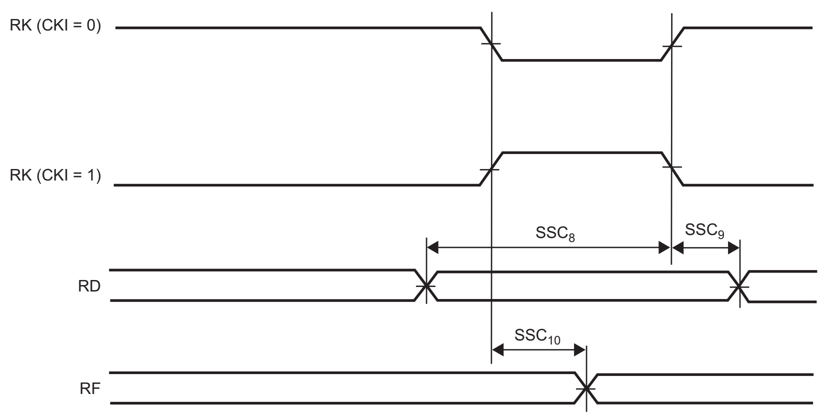


Figure 74-31. SSC Receiver, RK and RF in Output

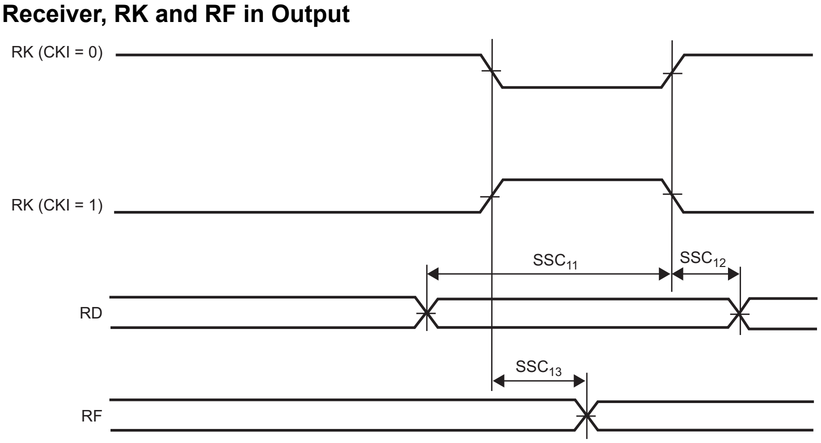


Figure 74-32. SSC Receiver, RK in Output and RF in Input

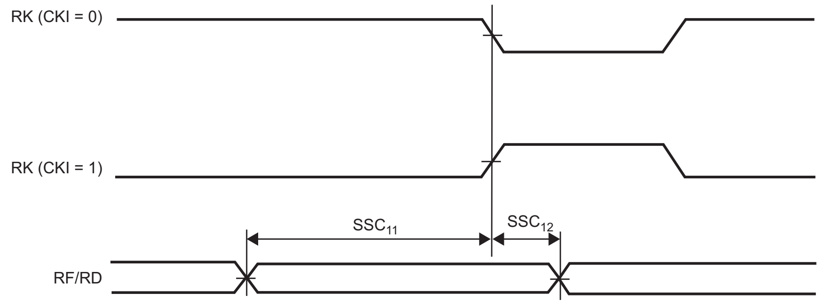


Table 74-33. SSC Timings

Symbol	Parameter	Conditions	Min	Max	Unit
		Transmitter			
f_{TK}	Frequency in Host or Client mode	–	–	25	MHz
SSC_0	TK edge to TF/TD (TK output, TF output)	–	0	8	ns
SSC_1	TK edge to TF/TD (TK input, TF output)	–	5	18	ns
SSC_2	TF setup time before TK edge (TK output)	–	15	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input)	STTDLY = 0 START = 4, 5 or 7	$2 \times t_{CPMCK}$	$8 + (2 \times t_{CPMCK})$	ns
SSC_5	TF setup time before TK edge (TK input)	–	0	–	ns
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	ns
SSC_7	TK edge to TF/TD (TK input, TF input)	STTDLY = 0 START = 4, 5 or 7	$5 + (3 \times t_{CPMCK})$	$18 + (3 \times t_{CPMCK})$	ns
SSC_{14}	TK low or high time[1]	–	–	10	ns
		Receiver			
f_{RK}	Frequency in Host or Client mode	–	–	25	MHz
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	ns
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input)	–	5	18	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	$7 - t_{CPMCK}$	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output)	–	0	8	ns

Note:

1. SSC_{14} and SSC_{15} apply to RK when RK is selected instead of TK (SSC_TCMR.CKS = RK).

74.5.12 I2SMCC Timings

Timings are provided in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 15 pF, DRV= 1, SR = 1

- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 15 pF, DRV= 0, SR = 1

Figure 74-33. I2SMCC Timing Diagram in Host Mode

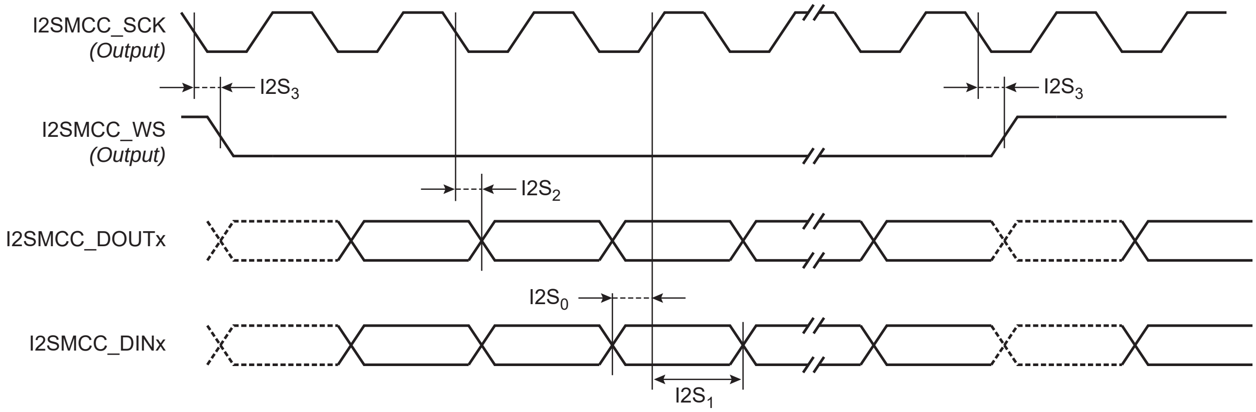


Figure 74-34. I2SMCC Timing Diagram in Client Mode

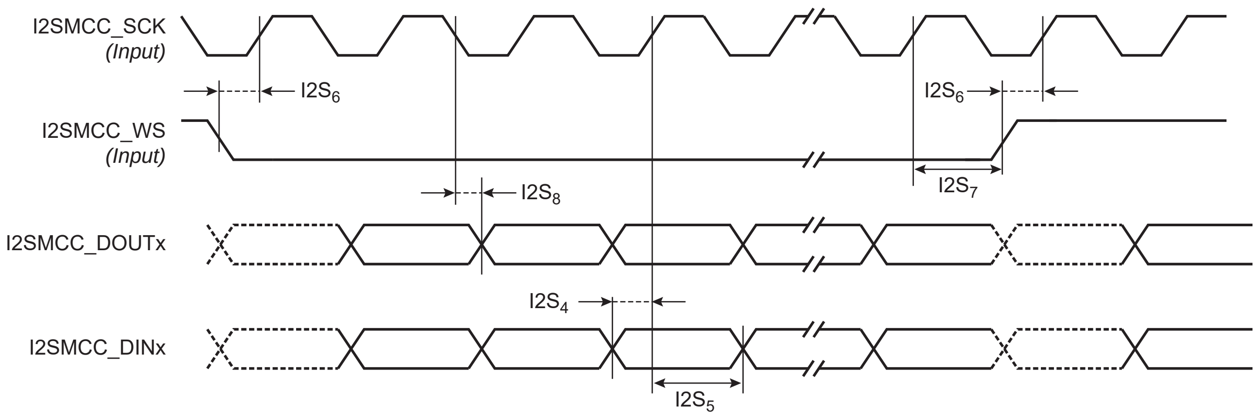


Table 74-34. I2SMCC Timings

Symbol	Parameter	Min	Max	Unit
Host Mode				
f_{I2SMCC_SCK}	Frequency in Host mode	–	25	MHz
$I2S_0$	I2SMCC_DINx setup time before I2SMCC_SCK rises	18	–	ns
$I2S_1$	I2SMCC_DINx hold time after I2SMCC_SCK rises	0	–	ns
$I2S_2$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	0	7	ns
$I2S_3$	I2SMCC_SCK falling to I2SMCC_WS delay	0	7	ns
Client Mode				
f_{I2SMCC_SCK}	Frequency in Client mode	–	25	MHz
$I2S_4$	I2SMCC_DINx setup time before I2SMCC_SCK rises	3	–	ns
$I2S_5$	I2SMCC_DINx hold time after I2SMCC_SCK rises	1	–	ns
$I2S_6$	I2SMCC_WS setup time before I2SMCC_SCK rises	7	–	ns
$I2S_7$	I2SMCC_WS hold time after I2SMCC_SCK rises	0	–	ns
$I2S_8$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	5	18	ns

74.5.13 ISC Timings

Timings are provided in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.00V to 3.6V, DRV = 0, SR = 1

Figure 74-35. ISC Timing Diagram

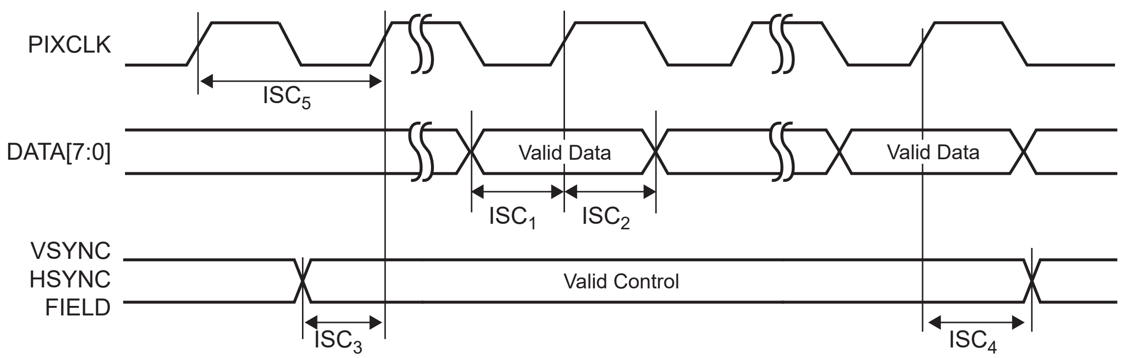


Table 74-35. ISC Timings

Symbol	Parameter	Min	Max	Unit
ISC_1	DATA setup time before PIXCLK rises	1	–	ns
ISC_2	DATA hold time after PIXCLK rises	0	–	ns
ISC_3	VSYNC/HSYNC/FIELD setup time before PIXCLK rises	1	–	ns
ISC_4	VSYNC/HSYNC/FIELD hold time after PIXCLK rises	0	–	ns
ISC_5	PIXCLK frequency	–	100	MHz

74.5.14 GMAC Timings

74.5.14.1 Ethernet MAC MDIO Mode

In MDIO mode, the MDC frequency must be less than or equal to 2.5 MHz to be compatible with the IEEE 802.3 MII specification.

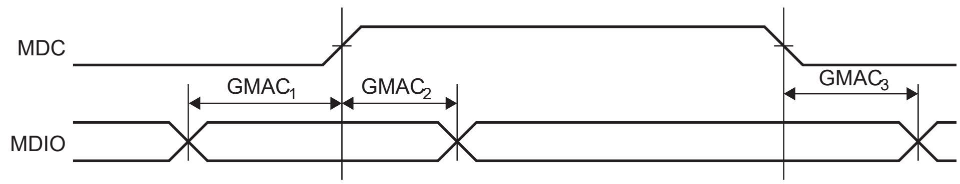
Timings are provided in the following conditions:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV= 1, SR = 1
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV= 0, SR = 1

Table 74-36. Ethernet MAC MDIO Interface Timings

Symbol	Parameter	Min	Max	Unit
$GMAC_1$	MDIO input data setup time before MDC rising edge	10	–	ns
$GMAC_2$	MDIO input data hold time after MDC rising edge	10	–	ns
$GMAC_3$	MDC falling edge to MDIO output data valid	0	300	ns

Figure 74-36. Ethernet MAC MDIO Interface Timings



74.5.14.2 Ethernet MAC MII Mode

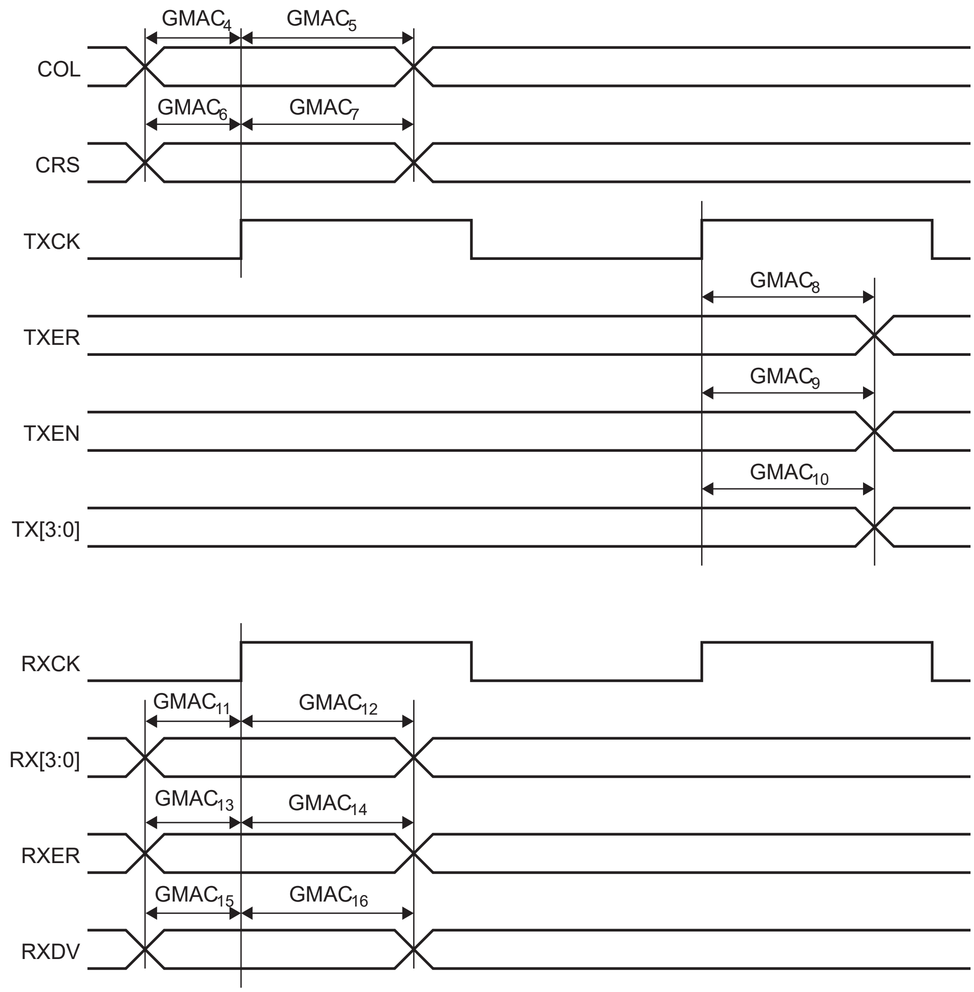
Timings are provided in the following conditions:

- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV= 1, SR = 1
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV= 0, SR = 1

Table 74-37. Ethernet MAC MII Specific Signals

Symbol	Parameter	Min	Max	Unit
$GMAC_4$	Setup for COL from TXCK rising	10	–	ns
$GMAC_5$	Hold for COL from TXCK rising	10	–	ns
$GMAC_6$	Setup for CRS from TXCK rising	10	–	ns
$GMAC_7$	Hold for CRS from TXCK rising	10	–	ns
$GMAC_8$	TXER toggling from TXCK rising	10	25	ns
$GMAC_9$	TXEN toggling from TXCK rising	10	25	ns
$GMAC_{10}$	TX[3:0] toggling from TXCK rising	10	25	ns
$GMAC_{11}$	Setup for RX[3:0] from RXCK	10	–	ns
$GMAC_{12}$	Hold for RX[3:0] from RXCK	10	–	ns
$GMAC_{13}$	Setup for RXER from RXCK	10	–	ns
$GMAC_{14}$	Hold for RXER from RXCK	10	–	ns
$GMAC_{15}$	Setup for RXDV from RXCK	10	–	ns
$GMAC_{16}$	Hold for RXDV from RXCK	10	–	ns

Figure 74-37. Ethernet MAC MII Mode



74.5.14.3 Ethernet MAC RMII Mode

Timings are provided in the following conditions:

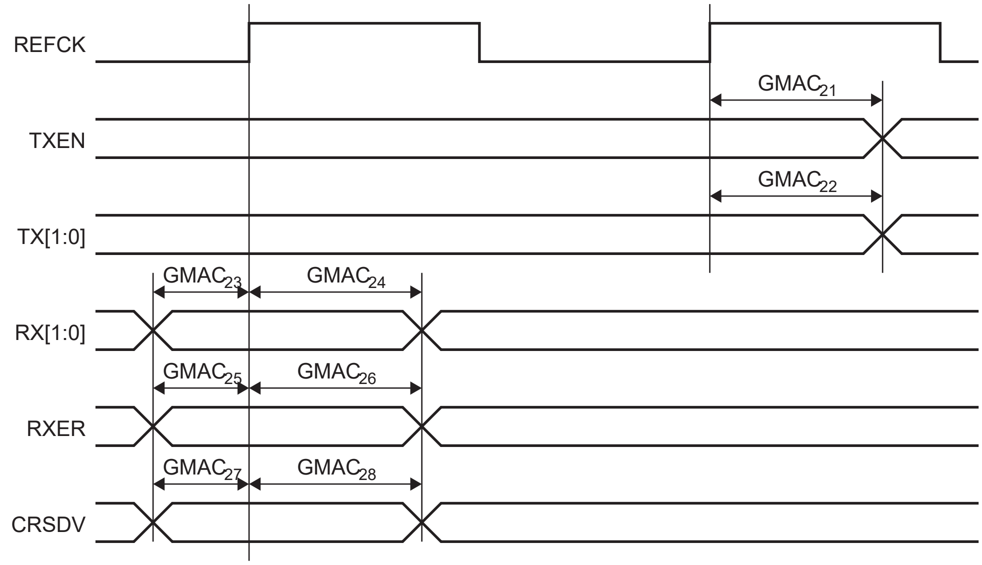
- 1.8V domain: VDDIO from 1.7V to 1.9V, maximum external capacitor = 10 pF, DRV= 1, SR = 0
- 3.3V domain: VDDIO from 3.0V to 3.6V, maximum external capacitor = 10 pF, DRV= 0, SR = 0

Table 74-38. Ethernet MAC RMII Mode

Symbol	Parameter	Min	Max	Unit
$GMAC_{21}$	TXEN toggling from REFCK rising	2	16	ns
$GMAC_{22}$	TX[1:0] toggling from REFCK rising	2	16	ns
$GMAC_{23}$	Setup for RX[1:0] from REFCK rising	4	–	ns
$GMAC_{24}$	Hold for RX[1:0] from REFCK rising	2	–	ns
$GMAC_{25}$	Setup for RXER from REFCK rising	4	–	ns
$GMAC_{26}$	Hold for RXER from REFCK rising	2	–	ns
$GMAC_{27}$	Setup for CRSDV from REFCK rising	4	–	ns

...........continued				
Symbol	**Parameter**	**Min**	**Max**	**Unit**
$GMAC_{28}$	Hold for CRSDV from REFCK rising	2	–	ns

Figure 74-38. Ethernet MAC RMII Timings

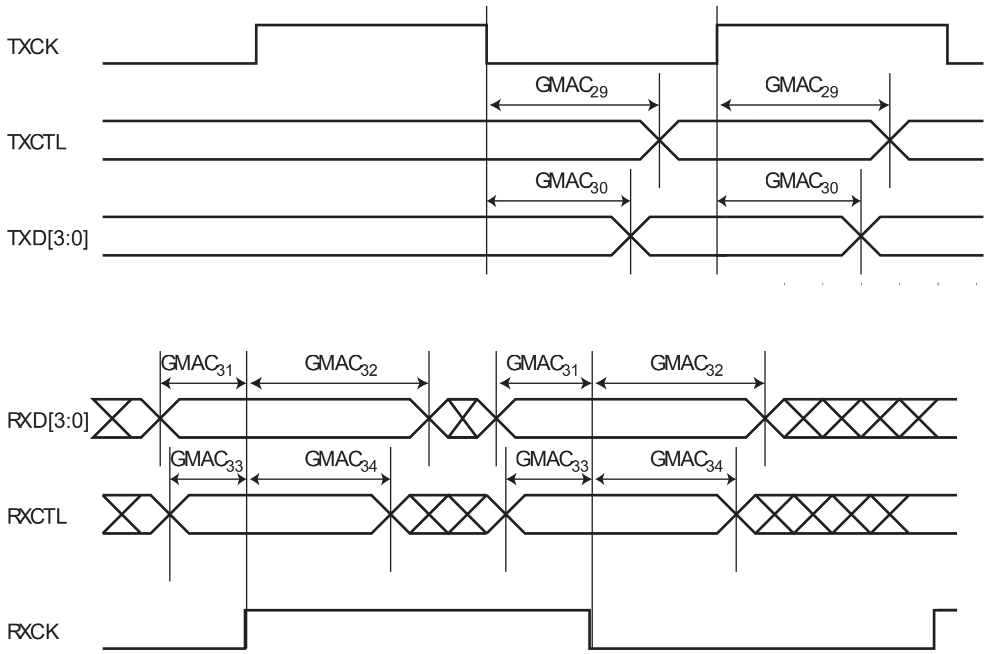


74.5.14.4 Ethernet MAC RGMII Mode

Table 74-39. RGMII V2.0 Specification

Symbol	Parameter	Min	Max	Unit
$GMAC_{29}$	TXCTL toggling from TXCK rising or falling edge	-500	500	ns
$GMAC_{30}$	TXD[3:0] toggling from TXCK rising or falling edge	-500	500	ps
$GMAC_{31}$	RXD[3:0] Setup time before RXCK rising or falling edge	1.0	–	ns
$GMAC_{32}$	RXD[3:0] Hold time after RXCK rising or falling edge	1.0	–	ns
$GMAC_{33}$	RXCTL Setup time before RXCK rising or falling edge	1.0	–	ns
$GMAC_{34}$	RXCTL Hold time after RXCK rising or falling edge	1.0	–	ns

Figure 74-39. RGMII Mode Timings



74.6 Embedded Analog Peripherals Characteristics

74.6.1 VDDOUT25 Voltage Regulator

Table 74-40. VDDOUT25 Voltage Regulator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	3.0	3.6	V
V_{START}	Input start-up voltage	–	V_{IT+} of POR VDDIN33		V
V_{STOP}	Input shutdown voltage	–	V_{IT-} of POR VDDIN33		V
t_{START}	Start-up time(1)	From VDDIN33 > V_{START} to VDDOUT25 set to 95% of its final value	–	6	ms
I_{LOAD}	Current consumption	Low-power mode (SFRBU_LDO25CR.LOWPOWER = 1)	–	5	mA
I_{LOAD_EXT}	External DC output current(2)	–	–	1	mA
$V_{DDOUT25}$	VDDOUT25 accuracy	–	2.45	2.55	V
I_{INRUSH}	Inrush current(1)(3)	I_{LOAD} = 0	–	100	mA
C_{IN}	Input decoupling capacitor(4)	–	2.2	–	μF
C_{OUT}	Stable output capacitor range(5)	Capacitance	1	2.7	μF
		ESR	0.01	0.3	Ω
R_{DIS}	Output discharge resistor value(1)	Regulator off	100	300	Ω

Notes:
1. Simulation data
2. This regulator is designed to supply the device internal circuits (PLLs, MIPI PHY, USB PHYs, oscillators, etc.). To supply external components, only DC current is permissible (for example, a resistive divider).
3. Input current when charging the external output capacitor C_{OUT}.
4. An X5R or X7R ceramic capacitor connected between VDDIN33 and the device's closest GND pin is a minimum requirement to reduce the inrush current and maximize the regulator performances.
5. To ensure stability, an external X5R or X7R ceramic output capacitor, C_{OUT}, must be connected between the VDDOUT25 and the device's closest GND pin.

74.6.2 VDDCORE Power-On Reset

Table 74-41. Core Power Supply POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDCORE}	Supply voltage range (VDDCORE)	–	0.5	1.21	V
V_{IT-_POR}	Negative-going input threshold voltage (VDDCORE)(1)	Measured with a -10V/s ramp rate	0.68	0.92	V
V_{IT+_POR}	Positive-going input threshold voltage (VDDCORE)(1)	Measured with a +10V/s ramp rate	0.70	0.99	V
V_{hys}	Hysteresis voltage(1)(2)	–	20	60	mV
t_{RES}	Reset time(1)	–	1	5	ms
t_{DET-}	V_{IT-} detection propagation time(1)	100 mV threshold overdrive	–	10	μs

Notes:
1. Simulation data
2. $V_{hys} = V_{IT+} - V_{IT-}$

74.6.3 VDDCPU Power-On Reset

Table 74-42. CPU Power Supply POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDCPU}	Supply voltage range (VDDCPU)	–	0.5	1.30	V
V_{IT-}	Negative-going input threshold voltage (VDDCPU)(1)	Measured with a -10V/s ramp rate	0.68	0.92	V
V_{IT+}	Positive-going input threshold voltage (VDDCPU)(1)	Measured with a +10V/s ramp rate	0.70	0.99	V
V_{hys}	Hysteresis voltage(1)(2)	–	20	60	mV
t_{RES}	Reset time(1)	–	1	5	ms
t_{DET-}	V_{IT-} detection propagation time(1)	100 mV threshold overdrive	–	10	μs

Notes:
1. Simulation data
2. $V_{hys} = V_{IT+} - V_{IT-}$

74.6.4 VDDIN33 Power-On Reset

Table 74-43. VDDIN33 POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	0.6	3.6	V
V_{IT-}	Negative-going input threshold voltage (VDDIN33)[(1)]	Measured with a -10V/s ramp rate	2.40	2.60	V
V_{IT+}	Positive-going input threshold voltage (VDDIN33)[(1)]	Measured with a +10V/s ramp rate	2.45	2.65	V
V_{hys}	Hysteresis voltage[(1)(2)]	–	40	55	mV
t_{RES}	Reset time[(1)]	–	1	6	ms
t_{DET-}	V_{IT-} detection propagation time[(1)]	100 mV threshold overdrive	–	30	µs

Notes:
1. Simulation data
2. $V_{hys} = V_{IT+} - V_{IT-}$

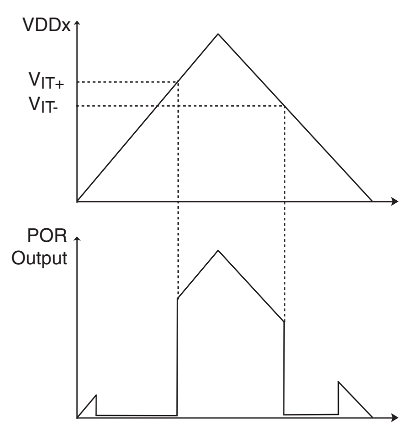
74.6.5 VBAT Power-On Reset

Table 74-44. VBAT POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{BAT}	Supply voltage range (VBAT)	–	0.6	3.6	V
V_{IT-}	Negative-going input threshold voltage (VBAT)[(1)]	Measured with a -10V/s ramp rate	1.40	1.57	V
V_{IT+}	Positive-going input threshold voltage (VBAT)[(1)]	Measured with a +10V/s ramp rate	1.42	1.59	V
V_{hys}	Hysteresis voltage[(1)(2)]	–	30	60	mV
t_{RES}	Reset time[(1)]	–	–	9	ms
t_{DET-}	V_{IT-} detection propagation time[(1)]	100 mV threshold overdrive	–	460	µs

Notes:
1. Simulation data
2. $V_{hys} = V_{IT+} - V_{IT-}$

Figure 74-40. VDDIN33/VDDCORE/VDDCPU/VBAT Power-On Reset Characteristics



74.6.6 Slow RC Oscillator

Table 74-45. Slow RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{BAT}	Supply voltage range (VBAT)[1]	–	V_{IT-}	3.6	V
f_{SLOWRC}	Frequency accuracy	VBAT = 3.3V, T_J = 0 to +50°C	30.5	33.5	kHz
		VBAT = V_{IT-} to 3.6V, over the applicable T_J range	29	35	kHz

Note:
1. In this table, V_{IT-} refers to the negative-going input threshold voltage of the VBAT POR (see the table VBAT POR Characteristics). The operation of the backup section of the device, and in particular of the slow RC oscillator, is granted down to the VBAT POR V_{IT-} threshold.

74.6.7 Main RC Oscillator

Table 74-46. Main RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33) [1]	–	3.0	3.6	V
I_{DDIN33}	Current consumption (VDDIN33)[2]	–	–	200	µA
t_{START}	Start-up time[2]	–	–	15	µs
f_0	Nominal output frequency		12		MHz
f_{ACC}	Output frequency accuracy	0°C < T_J < +50°C	–	±2	%
		-20°C < T_J < +70°C[2]	–	±3	%
		Over the applicable T_J range	–	±5	%
df/dV	Output frequency drift with VDDIN33[2]	V_{DDIN33}: 3.0V to 3.6V	–	0.01	% / V
Duty	Output duty cycle	–	45	55	%

Notes:
1. This oscillator is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data

74.6.8 32.768 kHz Crystal Oscillator

Table 74-47. 32.768 kHz Crystal Oscillator Characteristics

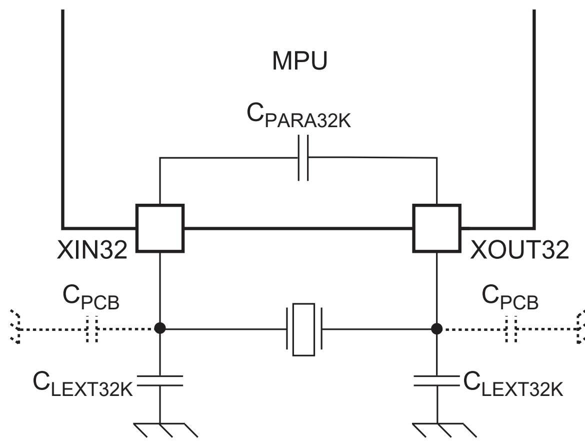
Symbol	Parameter	Conditions		Min	Max	Unit
V_{BAT}	Supply voltage range (VBAT)[1]	–		V_{IT-}	3.6	V
I_{VBAT}	Current consumption (VBAT)[2]	–		–	3.1	µA
t_{START}	Start-up time[2]	R_S[3] < 50 kΩ	C_M = 0.6fF	–	3.7	s
			C_M = 3fF	–	0.8	s
		R_S[3] < 90 kΩ	C_M = 0.6fF	–	5.0	s
			C_M = 3fF	–	1.0	s
f_{OSC}	Operating frequency	–		32.768		kHz
Duty	Duty cycle[2]	–		40	60	%

..........continued					
Symbol	**Parameter**	**Conditions**	**Min**	**Max**	**Unit**
R_F	Internal resistor[(2)]	Between XIN32 and XOUT32	6		MΩ
$C_{PARA32K}$	Internal parasitic capacitance[(2)]	Between XIN32 and XOUT32	2		pF

Notes:

1. In this table, V_{IT-} refers to the negative-going input threshold voltage of the VBAT POR (see the table VBAT POR Characteristics). The operation of the backup section of the device is granted down to the VBAT POR V_{IT-} threshold.
2. Simulation data
3. R_S is the crystal's equivalent series resistor.

Figure 74-41. 32.768 kHz Crystal Oscillator



$C_{LEXT32K} = 2x(C_{CRYSTAL} – C_{PARA32K} – C_{PCB} / 2)$

where C_{PCB} is the single-ended (ground-referenced) parasitic capacitance of the printed circuit board (PCB) on XIN32 and XOUT32 tracks. As an example, if the crystal is specified for a 12.5 pF load, with C_{PCB} = 1 pF (on XIN32 and on XOUT32), $C_{LEXT32K}$ = 2 x (12.5 - 2 - 0.5) = 20 pF.

The following table summarizes recommendations for 32.768 kHz crystal selection.

Table 74-48. Recommended 32.768 kHz Crystal Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	6	12.5	pF
R_S	Equivalent series resistor	–	–	90	kΩ
C_M	Motional capacitance	–	0.6	3	fF
C_{SHUNT}	Shunt capacitance	–	0.6	2	pF
P_{ON}	Drive level	–	0.2	–	µW

74.6.9 Main Crystal Oscillator

Table 74-49. Main Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)[(1)]	–	3.0	3.6	V
$I_{VDDIN33}$	Current consumption (VDDIN33)[(2)]	–	–	2.8	mA

...........continued

Symbol	Parameter	Conditions	Min	Max	Unit
t_{START}	Start-up time[(2)]		–	10	ms
f_{OSC}	Operating frequency range	–	12	48	MHz
Duty	Duty cycle[(2)]	–	40	60	%
C_{PARA}	Internal parasitic capacitance[(2)]	Between XIN and XOUT	1		pF

Notes:

1. This oscillator is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data

Three sets of crystal characteristics are supported with this oscillator corresponding to three operating frequency ranges:

- Set 1: Crystal frequency is between 12 and 20 MHz. See Table 74-50.
- Set 2: Crystal frequency is between 20 and 30 MHz. See Table 74-51.
- Set 3: Crystal frequency is between 30 and 48 MHz. See Table 74-52.

When choosing a crystal, one and only one of these sets must be completely satisfied.

Table 74-50. Recommended Crystal Characteristics (Set 1)

Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	12	20	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	17.5	pF
C_{SHUNT}	Shunt capacitance	–	–	3	pF
ESR	Equivalent series resistance	–	–	100	Ω
C_M	Motional capacitance	–	1.3	3.2	fF
P_{ON}	Drive level	–	150	–	µW

Table 74-51. Recommended Crystal Characteristics (Set 2)

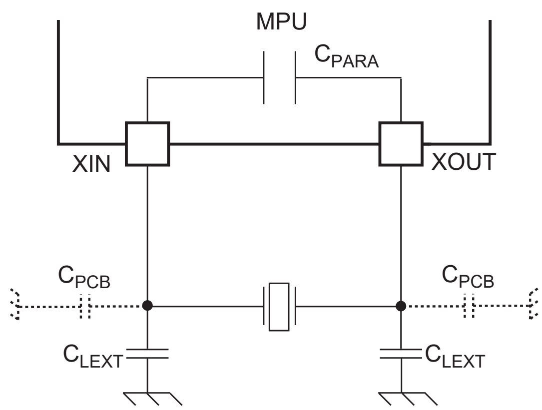
Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	20	30	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	12.5	pF
C_{SHUNT}	Shunt capacitance	–	–	3	pF
ESR	Equivalent series resistance	–	–	100	Ω
C_M	Motional capacitance	–	1.3	3.2	fF
P_{ON}	Drive level	–	300	–	µW

Table 74-52. Recommended Crystal Characteristics (Set 3)

Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	30	48	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	10	pF

...........continued					
Symbol	**Parameter**	**Conditions**	**Min**	**Max**	**Unit**
C_{SHUNT}	Shunt capacitance	With ESR_{MAX} = 60Ω	–	3	pF
		With ESR_{MAX} = 80Ω	–	1	pF
ESR	Equivalent series resistance	With C_{SHUNT_MAX} = 1 pF	–	80	Ω
		With C_{SHUNT_MAX} = 3 pF	–	60	Ω
C_M	Motional capacitance	–	1.3	3.2	fF
P_{ON}	Drive level	–	400	–	μW

Figure 74-42. Main Crystal Oscillator



C_{LEXT} = 2 x ($C_{CRYSTAL}$ – C_{PARA} – C_{PCB} / 2).

where C_{PCB} is the single-ended (ground referenced) parasitic capacitance of the printed circuit board (PCB) on XIN and XOUT tracks. As an example, if the crystal is specified for a 12.5 pF load, with C_{PCB}=1 pF (on XIN and on XOUT), C_{LEXT} = 2 x (12.5 - 1 - 0.5) = 22 pF.

74.6.10 Crystal Oscillator Design Considerations

When choosing a crystal for the 32.768 kHz Crystal Oscillator or for the Main Crystal Oscillator, several parameters must be taken into account. Important parameters are as follows:

- Crystal Load Capacitance: the total capacitance loading the crystal, including the oscillator's internal parasitics and the PCB parasitics, must match the load capacitance for which the crystal's frequency is specified. Any mismatch in the load capacitance with respect to the crystal's specification will lead to inaccurate oscillation frequency.
- Crystal Drive Level: use only crystals with specified drive levels greater than the minimum recommended value. Applications that do not respect this criterion may damage the crystal.
- Crystal Equivalent Series Resistance (ESR): use only crystals with a specified ESR lower than the maximum specified value. In applications where this criterion is not respected, the crystal oscillator may not start.
- Crystal Shunt Capacitance: use only crystals with a specified shunt capacitance lower than the maximum specified value. In applications where this criterion is not respected, the crystal oscillator may not start.
- PCB Layout Considerations: to minimize inductive and capacitive parasitics associated with XIN, XOUT, XIN32, XOUT32 nets, it is recommended to route them as short as possible. It is also of prime importance to keep those nets away from noisy switching signals (clock, data, PWM, etc.). A good practice is to shield them with a quiet ground net to avoid coupling to neighboring signals.

74.6.11 PLL Characteristics

The following characteristics apply to CPUPLL, DDRPLL, IMGPLL, SYSPLL, BAUDPLL, AUDIOPLL and ETHPLL and are given for register PMC_PLL_ACR programmed to the recommended value 0x00070010.

Table 74-53. PLL Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33) [1]	–	3.0	3.6	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.12	1.21	V
$I_{VDDIN33}$	Current consumption (VDDIN33)[2]	$f_{COREPLLCK}$ = 1.0 GHz	–	2.9	mA
$I_{VDDCORE}$	Current consumption (VDDCORE)[2]		–	3.5	mA
t_{START}	Start-up time[2]	To reach 95% of target frequency	–	50	μs
f_{IN}	Input frequency range	–	10	50	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	1000	MHz
$f_{IOPLLCK}$	IOPLLCK[3] frequency range	–	–	100	MHz

Notes:
1. The PLLs are powered by the 2.5V regulated output, which is supplied from VDDIN33.
2. Simulation data
3. IOPLLCK is available on the AUDIOPLL only and corresponds to the AUDIOCLK pin.

Table 74-54. PLL Output Clocks Characteristics

Symbol	Parameter[1]	Conditions	Min	Max	Unit
$f_{CPUPLLCK}$	CPUPLLCK frequency range	–	–	1000	MHz
$f_{DDRPLLCK}$	DDRPLLCK frequency range	–	–	533	MHz
$f_{IMGPLLCK}$	IMGPLLCK frequency range	–	–	266	MHz
$f_{SYSPLLCK}$	SYSPLLCK frequency range	–	–	416	MHz
$f_{BAUDPLLCK}$	BAUDPLLCK frequency range	–	–	208	MHz
$f_{AUDIOPLLCK}$	AUDIOPLLCK frequency range	–	–	200	MHz
$f_{ETHPLLCK}$	ETHPLLCK frequency range	–	–	125	MHz

Note:
1. Simulation data

74.6.12 Temperature Sensor and Voltage Reference Characteristics

The temperature sensor provides two output voltages that can be converted by the internal 12-bit ADC:

- V_{TEMP}: proportional to the junction temperature (T_J), and
- V_{BG}: temperature-independent voltage.

Both voltages are multiplexed to ADC channel 31. The selection of VTEMP or VBG is done with the ADC_ACR.SRCLCH bit. Calibration data for the temperature sensor are factory-programmed in one OTP controller packet. These calibration data can only be read and used after writing a “Boot Configuration Packet” in the OTP memory. Refer to the application note “SAMA7G5 Series Temperature Sensor Calibration”, available on microchip.com.

Table 74-55. Temperature Sensor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)[1]	–	3.0	3.6	V
$I_{VDDIN33}$	Current consumption[2]	–	–	200	μA
t_{START}	Start-up time[3]	–	–	20	μs

...........continued

Symbol	Parameter	Conditions	Min	Max	Unit
V_{TEMP}	Temperature output voltage	T_J = 25°C	0.59	0.67	V
dV_{TEMP}/dT	V_{TEMP} sensitivity to temperature(2)	–	2.08		mV/°C
T_{ACC}	Temperature reading accuracy(2)(3)	Over the applicable T_J range	–	±5	°C
V_{BG0}	VBG reference output voltage	T_J = 27°C	1.19	1.21	V
ΔV_{BG}	VBG drift with temperature(2)	$\Delta V_{BG} = V_{BG}(T_J)-V_{BG0}$	-20	20	mV
t_{TH}	V_{TEMP} or V_{BG} track-and-hold time(2)	Required track-and-hold time to ensure 1 LSB accurate settling on internal 12-bit ADC	20	–	µs

Notes:
1. The temperature sensor is powered by the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data
3. Using the temperature sensor calibration procedure described in the application note “SAMA7G5 Series Temperature Sensor Calibration”, available on microchip.com.

74.6.13 12-bit ADC Characteristics

Table 74-56. ADC Power Supply and Voltage Reference Input Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.6	V
$I_{VDDIN33}$	Current consumption on VDDIN33(1)(2)	Low Speed – $f_S \le$ 500kS/s ADC_ACR.IBCTL = $(00)_2$	–	1.0	mA
		Full Speed – $f_S \le$ 1MS/s ADC_ACR.IBCTL = $(01)_2$	–	1.8	mA
$V_{ADVREFP}$	ADVREFP input voltage range	–	2.4	2.55	V
$R_{ADVREFP}$	ADVREFP input resistance to ground(2)	ADC off	7.2	12	kΩ
		ADC on	1	–	MΩ
$C_{ADVREFP}$	Recommended decoupling capacitor on ADVREFP	–	1	–	µF

Notes:
1. The 12-bit ADC is powered by the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data

Table 74-57. ADC Timing Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CKADC}	ADC clock frequency	ADC_ACR.IBCTL = $(00)_2$	0.1	10	MHz
		ADC_ACR.IBCTL = $(01)_2$	0.2	20	MHz
t_{CONV}	ADC conversion time(1)	–	20	–	t_{CKADC}
f_S	Sampling rate(2)	ADC_ACR.IBCTL = $(00)_2$	–	0.5	MS/s
		ADC_ACR.IBCTL = $(01)_2$	–	1	MS/s
t_{START}	Start-up time(3)	From Off to On	–	5	µs

...........continued					
Symbol	**Parameter**	**Conditions**	**Min**	**Max**	**Unit**
t_{TRACK}	Track and hold time(3)(4)	–	300	–	ns

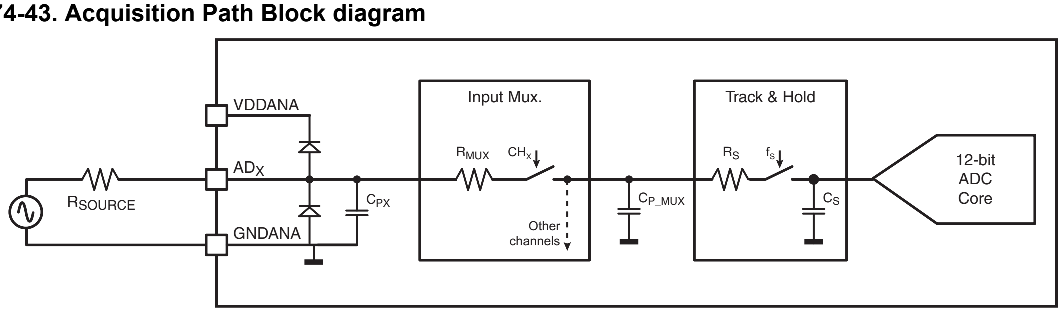
Notes:
1. $t_{CONV} = t_{CH} + t_{TRACK} + 14 \times t_{CKADC}$ with $t_{CKADC} = 1 / f_{CKADC}$. $t_{CH} = 0$ when the ADC operates in the same input mode (single-ended, pseudo-differential or differential) for the current conversion than for the previous one. $t_{CH} = 2$ when the ADC input mode is changed to perform the current conversion.
2. $f_S = 1 / t_{CONV}$
3. Simulation data
4. See Track and Hold Time versus Source Impedance – Sampling Rate.

Table 74-58. ADC Analog Input Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{FS}	Analog input full scale range(1)	ADC_CCR.DIFFx = 0	0	$V_{ADVREFP}$	V
		ADC_CCR.DIFFx = 1	$-V_{ADVREFP}$	$V_{ADVREFP}$	V
V_{INCM}	Common mode input range in Differential Input mode(2)	ADC_CCR.DIFFx = 1	$0.4 \times V_{DDANA}$	$0.6 \times V_{DDANA}$	V
C_S	ADC sampling capacitance(3)	–	–	3	pF
C_{P_ADx}	ADx input parasitic capacitance(3)(4)	ADx pin configured as analog input	–	7	pF
R_{ON}	Internal series resistor(3)(4)	–	–	2	kΩ
Z_{IN}	Common mode input impedance(3)(5)	On ADx pin	$1 / (f_S.C_S)$	–	Ω
R_{CH30}	VBAT resistive attenuator impedance	–	80	120	kΩ
G_{CH30}	VBAT channel gain	–	0.595	0.605	–

Notes:
1. $V_{FS} = (V_{ADx} - V_{GNDANA})$ in Single-ended mode, $V_{FS} = (V_{ADx} - V_{AD11})$ in Pseudo-differential mode, and $V_{FS} = (V_{ADx} - V_{ADx+1})$ in Differential mode
2. $V_{INCM} = (V_{ADx} + V_{ADx+1}) / 2$
3. Simulation data
4. With respect to the equivalent model of the figure Equivalent Model of the Acquisition Path
5. Assuming conversion on one single channel

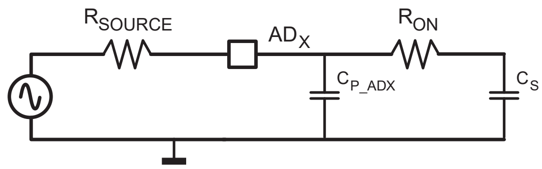
Figure 74-43. Acquisition Path Block diagram



For tracking time calculation, during the sampling phase of the converter, this acquisition path can be reduced to the equivalent model of the following figure, where:

- $R_{ON} = R_{MUX} + R_S$
- $C_{P\_ADX} = C_{PX} + C_{P\_MUX}$

Figure 74-44. Equivalent Model of the Acquisition Path



See Track and Hold Time versus Source Impedance – Sampling Rate for further details on the use of this model.

In the following table, unless otherwise specified, the specifications are provided for two speed operating ranges.

- Source resistance = 50 Ω
- ADC_EMR.OSR<2:0> = $(000)_2$
- Low-speed
  – f_{CKADC} = 10 MHz, f_S = 500 kS/s
  – ADC_ACR.IBCTL = $(00)_2$
- High-speed
  – f_{CKADC} = 20 MHz, f_S = 1 MS/s
  – ADC_ACR.IBCTL = $(01)_2$

Table 74-59. Static Performance Characteristics(1)

Symbol	Parameter	Min	Max	Unit
RES_{ADC}	Native ADC resolution	12		Bit
INL	Integral non-linearity	-3	3	LSB
DNL	Differential non-linearity	-2	2	LSB
OE	Offset error(2)	-4	4	LSB
GE	Gain error(2)	-4	4	LSB

Notes:
1. In this table, errors are expressed in LSB where:
   – LSB = $V_{VREFP} / 2^{12}$ in Single-ended mode (ADC_CCR.DIFFx = 0 and ADC_PDR.PDIFFx = 0)
   – LSB = $V_{VREFP} / 2^{11}$ in Differential or Pseudo-differential mode (ADC_CCR.DIFFx = 1)
2. Error with respect to the best fit line method.

74.6.13.1 Track and Hold Time versus Source Impedance - Sampling Rate

Referring to the figure Equivalent Model of the Acquisition Path, during its tracking phase, the 12-bit ADC charges its sampling capacitor C_S through various serial resistors modeled as R_{SOURCE} (source output resistor) and R_{ON} (multiplexer series resistor and the sampling switch series resistor). In case of high output source resistance (for example, a low power resistive divider), the tracking time must be increased to ensure full settling of the sampling capacitor voltage. Note that programming a long tracking time may impact the sampling frequency (f_S). The following formula provides the minimum tracking time that ensures a 12-bit accurate settling.

$t_{TRACK} \geq 8 \times (R_{SOURCE} + R_{ON}) \times C_S$

The ADC Controller (ADCC) counts the tracking time in ADC clock cycles (t_{CKADC}). This time can be adjusted between 6 and 54 cycles in the fields ADC_MR.TRACKTIM and ADC_EMR.TRACKX4. At maximum ADC clock frequency (20 MHz), the maximum tracking time that can be programmed is 2.7 µs. This limits 12-bit accurate sampling to sources having R_{SOURCE} in the 100 kΩ range. To overcome this limitation, the ADC clock frequency can be decreased.

The following examples show typical use cases of tracking time and sampling frequency calculation.

Example 1: Calculated tracking time is lower than the default (minimum) tracking time.

- Assuming f_{CKADC} = 8 MHz (t_{CKADC} = 125 ns), R_{SOURCE} = 10 kΩ

- The minimum required track time is t_{TRACK} = 8 x (10 kΩ + 2 kΩ) x 3pF = 288 ns
- t_{TRACK} is less than the minimum tracking time (6 x t_{CKADC} = 750 ns): set TRACKTIM = 0 and TRACKX4 = 0
- The real tracking time is 6 x t_{CKADC} (750 ns) and the conversion time is:
  t_{CONV} = t_{TRACK} + 14 x t_{CKADC}= 20 x t_{CKADC}
- The sampling rate is f_S = 8 MHz / 20 = 400 kS/s
- The maximum allowable source resistance is R_{SOURCE_MAX} = (6 x t_{CKADC}) / (3 pF x 8) - 2 kΩ= 29.25 kΩ

Example 2: Calculated tracking time is greater than the default (minimum) tracking time.

- Assuming f_{CKADC} = 20 MHz (t_{CKADC} = 50 ns), R_{SOURCE} = 20 kΩ
- The minimum required track time is t_{TRACK} = 8 x (20 kΩ + 2 kΩ) x 3 pF= 528 ns
- t_{TRACK} is greater than the minimum tracking time (6 x t_{CKADC} = 300 ns): set TRACKTIM=5 and TRACKX4=1
- The real tracking time is (4 x (5 + 1) - 10) = 14 x t_{CK_ADC} = 700 ns
- The conversion time is t_{CONV} = t_{TRACK} + 14 x t_{CKADC}= 28 x t_{CKADC}
- The sampling rate is f_S = 20 MHz / 28 = 714.3 kS/s
- The maximum allowable source resistance is R_{SOURCE_MAX} = (14 x t_{CKADC}) / (3 pF x 8) - 2 kΩ= 27.2 kΩ

Example 3: Maximum sampling rate operation.

- Assuming f_{CKADC} = 20 MHz (t_{CKADC} = 50 ns), R_{SOURCE} = 10 kΩ
- The minimum required track time is t_{TRACK} = 8 x (10 kΩ + 2 kΩ) x 3 pF= 288 ns
- t_{TRACK} is less than the minimum tracking time (6 x t_{CKADC} = 300 ns): set TRACKTIM=0 and TRACKX4=0
- The real tracking time is 6 x t_{CK_ADC} (300 ns) and the conversion time is:
  t_{CONV} = t_{TRACK} + 14 x t_{CKADC}= 20 x t_{CKADC}
- The sampling rate is f_S = 20 MHz / 20 = 1 MS/s
- The maximum allowable source resistance is R_{SOURCE_MAX} = (6 x t_{CKADC}) / (3 pF x 8) - 2 kΩ= 10.5 kΩ

74.6.14 Analog Comparator Characteristics

The Analog Comparator features a built-in hysteresis voltage that applies on positive-going differential voltages. The following definitions are used to list the comparator specifications:

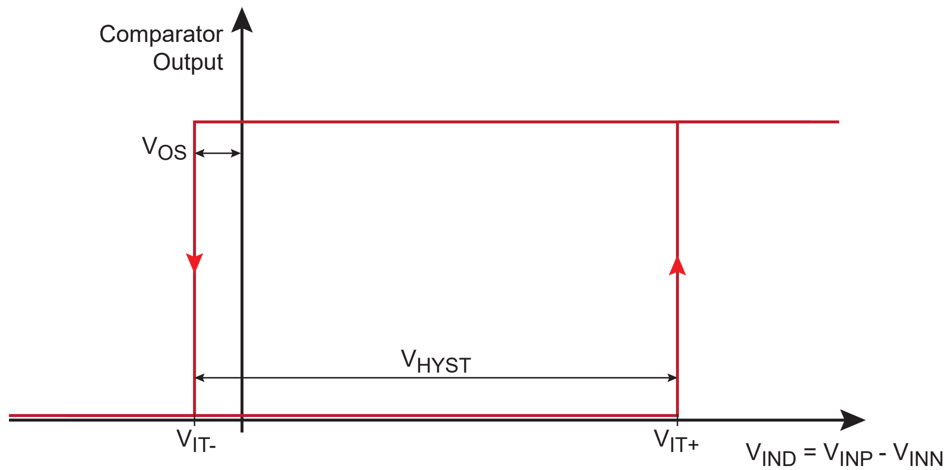
- V_{IND} = (V_{INP} - V_{INN}) is the differential input voltage
- V_{IT+} is the input threshold for a positive-going differential voltage
- V_{IT-} is the input threshold for a negative-going differential voltage

Table 74-60. Analog Comparator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{VDDIN33}$	Supply voltage range (VDDIN33)(1)	–	3.0	3.6	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	–	–	80	µA
t_{START}	Start-up time(2)	–	–	5	µs
t_{PD}	Propagation delay(2)(3)	–	–	200	ns
V_{CM}	Common mode input range	On negative and positive inputs	0.25	2.25	V
V_{OS}	Input offset voltage(2)	Applies to positive-going V_{IND}	-10	10	mV
V_{HYST}	Hysteresis voltage(2)(4)	–	–	40	mV

Notes:
1. The analog comparator is powered by the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data
3. Measured with 100 mV threshold overload
4. V_{HYST} = (V_{IT+} - V_{IT-})

Figure 74-45. Analog Comparator Input/Output Transfer Function



74.6.15 HS USB Transceiver Characteristics

The device complies with all voltage, power, and timing characteristics and specifications as set forth in the USB 2.0 Specification. Refer to the USB 2.0 Specification for more information.

Table 74-61. USB 2.0 Transceiver Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{VDD33}	Supply voltage range (VDDIN33 and VDDUTMII)[1]	–	3.0	3.6	V
I_{VDD33}	Current consumption in VDDIN33 and VDDUTMII[2]	HS TX	–	23	mA
		HS idle	–	7	mA
		LS/FS TX	–	18	mA
$I_{VDDCORE}$	Current consumption in VDDCORE[2]	HS TX	–	16	mA
		HS Idle	–	17	mA
		LS/FS TX	–	14	mA

Notes:
1. VDDIN33 and VDDUTMII are powered from one single power source so that ΔV(VDDIN33,VDDUTMII) ≤ 50 mV.
2. Simulation data

74.7 Security Module Characteristics

74.7.1 Environmental Sensors

74.7.1.1 Voltage Monitoring

Table 74-62. VDDCORE Voltage Monitor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)[1]	–	3.00	3.60	V

...........continued

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT+}	Positive-going input threshold voltage (VDDCORE overvoltage)	–	1.17	1.21	V
V_{IT-}	Negative-going input threshold voltage (VDDCORE undervoltage)	–	0.99	1.03	V
V_{hys}	Hysteresis voltage(2)(3)	–	30		mV
t_{DET}	Detection time(3)(4)	–	–	50	μs

Notes:
1. This voltage monitor is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Hysteresis applies to both VIT+ and VIT- voltage thresholds. VIT+ has a negative hysteresis and VIT- has a positive hysteresis. Refer to Security Module (SECUMOD).
3. Simulation data
4. An additional debouncing time applies to this detection time as programmed in SECUMOD.VDDCORE_DBTV.

Table 74-63. VDDCPU Voltage Monitor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.00	3.60	V
V_{IT+}	Positive-going input threshold voltage (VDDCPU overvoltage)	800 MHz operating frequency (SECUMOD_GPSBR.SMCPURANGE = 1)	1.26	1.30	V
		600 MHz operating frequency (SECUMOD_GPSBR.SMCPURANGE = 0)	1.17	1.21	V
V_{IT-}	Negative-going input threshold voltage (VDDCPU undervoltage)	800 MHz operating frequency (SECUMOD_GPSBR.SMCPURANGE = 1)	1.10	1.14	V
		600 MHz operating frequency (SECUMOD_GPSBR.SMCPURANGE = 0)	0.99	1.03	V
V_{hys}	Hysteresis voltage(2)(3)	–	30		mV
t_{DET}	Detection time(3)(4)	–	–	50	μs

Notes:
1. This voltage monitor is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Hysteresis applies to both VIT+ and VIT- voltage thresholds. VIT+ has a negative hysteresis and VIT- has a positive hysteresis. Refer to Security Module (SECUMOD).
3. Simulation data
4. An additional debouncing time applies to this detection time as programmed in SECUMOD_VCPUFR.VDDCPU_DBTV.

Table 74-64. VBAT Voltage Monitor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{BAT}	Supply voltage range (VBAT)(1)	–	V_{IT-_POR}	3.60	V
I_{BAT}	Current consumption on VBAT(2)(3)	–	–	200	nA
V_{IT+}	Positive-going input threshold voltage (VBAT overvoltage)	Scheduled operation every 20 ms	3.41	3.64	V
V_{IT-}	Negative-going input threshold voltage (VBAT undervoltage)	Scheduled operation every 20 ms	1.49	1.64	V
t_{DET}	Detection time(3)(4)	–	–	10	μs

Notes:
1. V_{IT-_POR} refers to the negative-going input threshold of the VBAT power-on reset (see the table VBAT POR Characteristics). Operation of the device backup section, and in particular of this monitor, is granted down to the VBAT power-on reset V_{IT-} threshold.
2. Average current consumption. Refer to Security Module (SECUMOD).
3. Simulation data
4. Measured with 100 mV threshold overload

Table 74-65. VDDANA Voltage Monitor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	3.00	3.60	V
I_{DDIN33}	Current consumption (VDDIN33)(1)	Normal mode	–	120	µA
		Low-power mode	–	80	µA
V_{IT+}	Positive-going input threshold voltage (VDDANA overvoltage)(1)	–	2.575	2.675	V
V_{IT-}	Negative-going input threshold voltage (VDDANA undervoltage)(1)	–	2.325	2.425	V
t_{DET}	Detection time(1)	–	–	50	µs

Note:
1. Simulation data

74.7.1.2 Temperature Monitoring

Table 74-66. Temperature Monitor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{BAT}	Supply voltage range (VBAT)(1)	–	V_{IT-_POR}	3.60	V
I_{BAT}	Current consumption on VBAT(2)(3)	–	–	200	nA
T_{JT+}	Positive-going junction temperature threshold(3)	High threshold (SECUMOD_GPSBR.TSRANGE = 1 or set by OTP)	110	125	°C
		Low threshold (SECUMOD_GPSBR.TSRANGE = 0 or set by OTP)	95	110	°C
T_{JT-}	Negative-going junction temperature threshold(3)	–	-40	-30	°C

Notes:
1. V_{IT-_POR} refers to the negative-going input threshold of the VBAT Power-on-Reset (See the table VBAT POR Characteristics). Operation of the device backup section, and in particular of this monitor, is granted down to the VBAT Power-On-Reset V_{IT-} threshold.
2. Average current consumption
3. Simulation data

74.8 Power Consumption in Active Mode

Tables Processor Power Consumption running a Coremark Benchmark from SRAM on AMP1/AMP2 and Use Case Definiton report active power consumption data measured on a few SAMA7G5 typical process samples. These data do not provide maximum power consumption specifications.

74.8.1 Processor Power Consumption in Active Mode

The following table provides the processor power consumption in the following conditions:

- f_{CPU_CLK} = from 400 MHz to 1000 MHz

- f_{MCK1} = 200 MHz
- f_{MCK2} = 533 MHz
- f_{MCK3} = 266 MHz
- f_{MCK4} = 400 MHz
- L1 caches enabled
- L2 cache enabled
- The Cortex-A7 core executes a CoreMark benchmark from the (internal) SRAM
- Code compiled with speed optimization
- Peripheral clocks disabled
- Current measured according to the following figure

Figure 74-46. Current Measurement on VDDCORE and VDDCPU

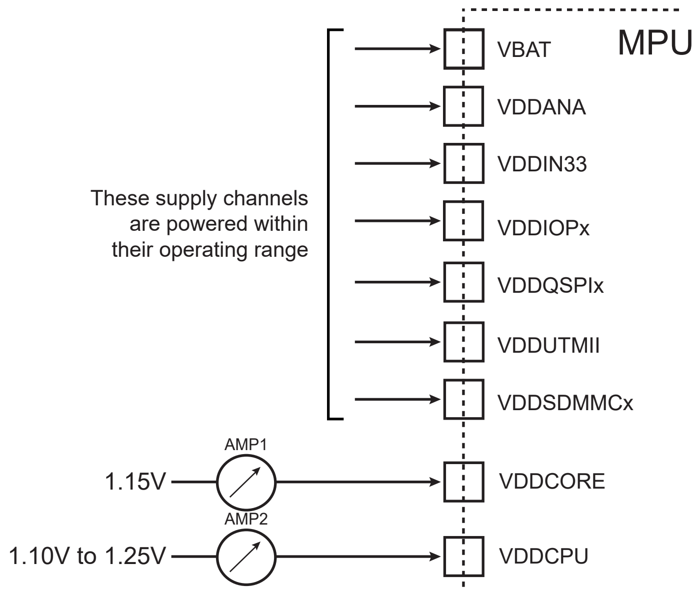
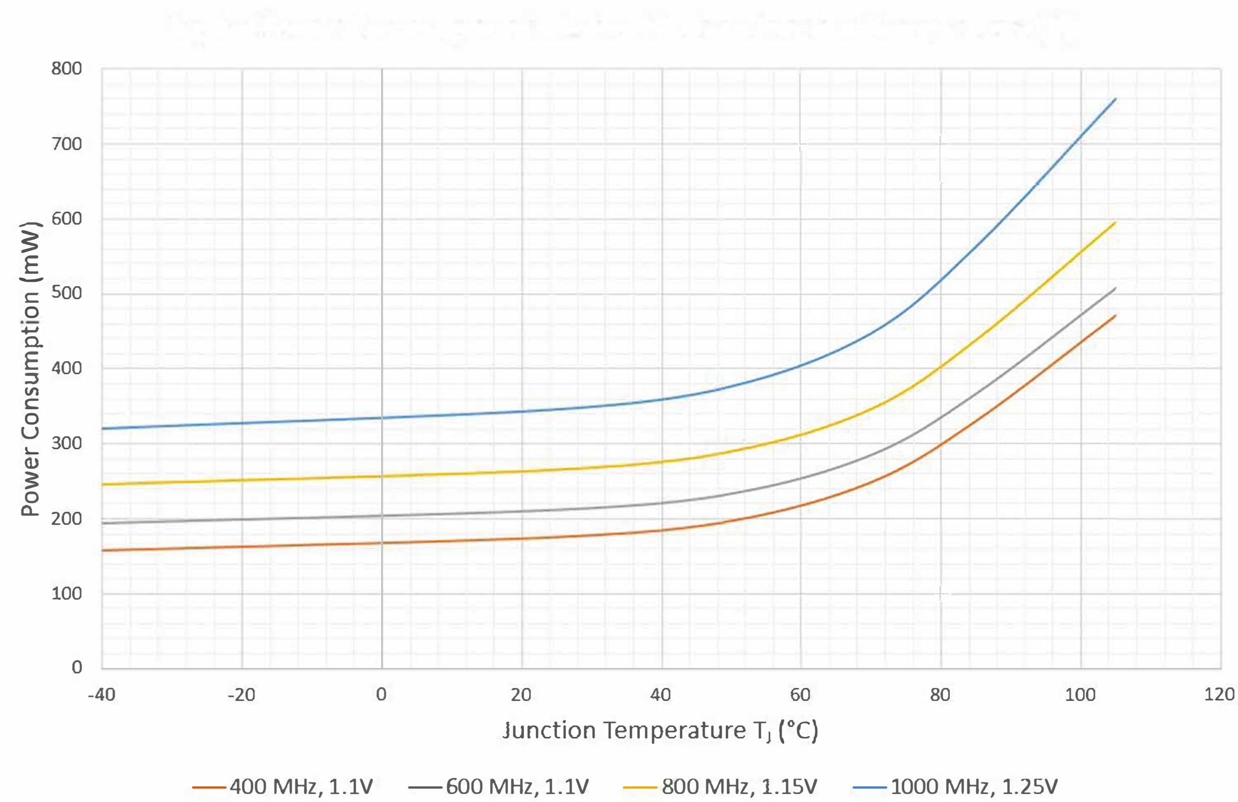


Table 74-67. Processor Power Consumption running a Coremark Benchmark from SRAM on AMP1+AMP2

f_{CPU} (MHz)	VDDCPU (V)	Current (mA)	T_J (°C)					
			-40	25	50	70	85	105
400	1.1	IDDCORE	82	89	98	119	153	211
		IDDCPU	57	66	76	101	140	208
600	1.1	IDDCORE	82	89	98	119	153	211
		IDDCPU	90	99	109	134	173	241
800	1.15	IDDCORE	82	89	98	119	153	211
		IDDCPU	131	141	153	181	227	306
1000	1.25	IDDCORE	82	89	98	119	153	211
		IDDCPU	181	195	211	248	309	413

Figure 74-47. Typical Processor Power Consumption when Running a CoreMark Benchmark



74.8.2 System Power Consumption in Applicative Use Cases

The table Power Consumption in Applicative Use Cases provides the processor power consumption in the following conditions:

- f_{CPU_CLK} = 800 MHz
- f_{MCK1} = 200 MHz
- f_{MCK2} = 533 MHz
- f_{MCK3} = 266 MHz
- f_{MCK4} = 400 MHz
- I & D caches are enabled.
- The external SDRAM memory is DDR3L.
- Use cases are run on Linux.
- Current consumptions are measured according to the figure below. Note that the external component current consumptions are not counted.
- The ambient temperature is 25°C.

Table 74-68. Use Case Definition

Use Case	Description
1	Audio MP3 decoding and playback on I^2S; MP3 file on USB mass storage
2	SAMA7G5 running as iPerf server
3	Running Bonnie++ on USB mass storage
4	SAMA7G5 downloads a file from GMAC0 and copies this file to USB mass storage
5	Streaming camera on Ethernet (image format RAW8, 1080p @ 30fps)

Figure 74-48. Current Measurement for Applicative Use Cases

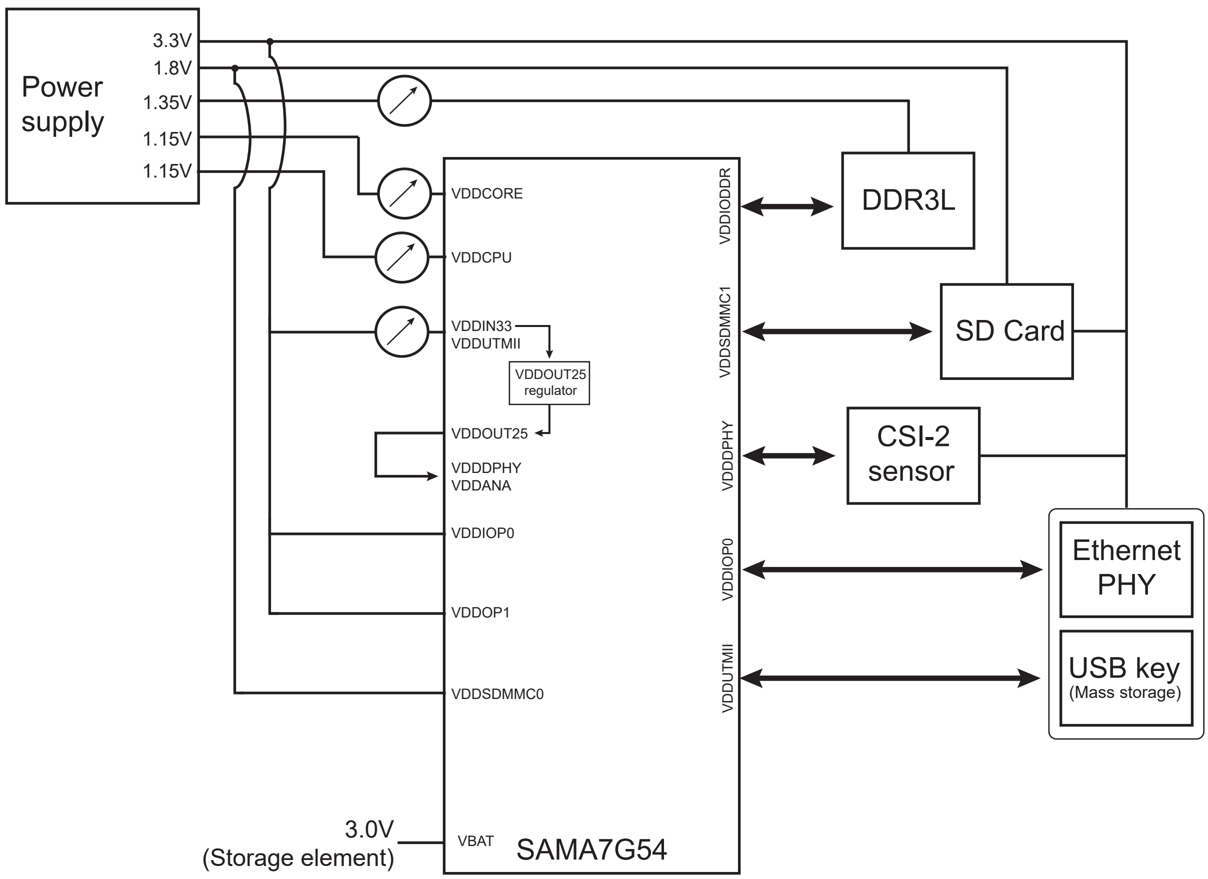


Table 74-69. Power Consumption in Applicative Use Cases

Use Case	Power Consumption (mW)				
	VDDCORE 1.15V	VDDCPU 1.15V	VDDIODDR 1.35V	VDDIN33 VDDUTMII 3.3V	Total
1	283	59	53	127	522
2	318	149	111	157	735
3	310	175	66	123	674
4	315	150	98	121	684
5	320	127	82	138	667

74.9 Operation and Power Consumption in Low-Power Modes

The SAMA7G5 features six low-power modes summarized in the following table. A detailed description of each mode is provided in the following sections.

Table 74-70. Low-Power Modes Summary

		Low-Power Mode					
		Backup Mode	BSR	ULP2	ULP1	ULP0	Idle
VBAT Power		✓	✓	✓	✓	✓	✓
VDDCPU Power		✗	✗	✗	✓	✓	✓
VDDCORE Power		✗	✗	✓	✓	✓	✓
SDRAM or DDR Power		✗	✓	✓	✓	✓	✓
DDR Self-refresh		✗	✓	✓	✓	✓	✗
Main Crystal Oscillator		✓	✓	✓	✓	✓	✓
Main RC Oscillator		✓	✓	✓	✓	✓	✓
Main Clock (MAINCK)	Source (MOSCSEL)	–	–	Main RC osc.	Main RC osc.	Main RC osc.	User-defined
	Status	–	–	✗	✗	✗	✓
Main System Bus Clock (MCK1)	Source (CSS)	–	–	MAINCK	MAINCK	MD_SLCK (divided by 64)	User-defined (PLLs, etc.)
	Status	–	–	✗	✗	✓	✓
CPU Clock (CPU_CLK)		–	–	✗	✗	✓	✓
Cortex-A7 State		Not powered	Not powered	Not powered	Clocks stopped	WFI	WFI
Mode Entry		Use SHDWC to shut down all power supplies except VBAT.	Context saved in SDRAM. SDRAM in Self-refresh. Use SHDWC to shut down all power supplies except VDDIODDR and VBAT.	Set AUTOLPM bit. Context saved in SDRAM. SDRAM in Self-refresh. ULP2 bit	Context saved in SDRAM. SDRAM in Self-refresh. ULP1 bit	Context saved in SDRAM. SDRAM in Self-refresh. WFI	WFI
Wake-up	Sources	RTC/RTT alarm. WKUP0, SECUMOD events	RTC/RTT alarm. WKUP0, SECUMOD events	Any PIO line. RTC/RTT alarm. USB resume.	Any PIO line. RTC/RTT alarm. USB resume.	Any unmasked interrupt	Any unmasked interrupt
	Event Sampling Clock	MD_SLCK	MD_SLCK	Asynchronous	Asynchronous	MD_SLCK	User-defined (MCKx, periph_clk, etc.)
Main System Bus Clock (MCK1) at Wake-up		MAINCK (defaults to Main RC osc.)	MAINCK (defaults to Main RC osc.)	MAINCK (configured to Main RC osc.)	MAINCK (configured to Main RC osc.)	MD_SLCK	User-defined
Wake-up Time (time to fetch first instruction)		-	-	Table 74-73			

Note:

1. ✓ / ✗ means either powered/un-powered for a power source, on/off for an oscillator, active/inactive for a clock signal, or enter/exit Self-Refresh mode for the external SDRAM.

74.9.1 Backup Mode

74.9.1.1 Operation

Backup mode is designed to serve prolonged power-down periods of the processor. In this mode, only the backup area of the device powered by VBAT is maintained (RTC, GPBR, SHDWC, BSCR, SECUMOD). This mode is entered by shutting down all the power rails of the device except VBAT. It is good practice to use the SHDN output of the Shutdown Controller (SHDWC) for this purpose.

The device exits Backup mode when an active wake-up event is triggered by the SHDWC. Upon this wake-up event, the SHDWC automatically toggles its SHDN output back to VBAT and this signal typically helps to restart all the power supply channels at board level.

The wake-up events to exit Backup mode are:

- WKUP0 pin
- SECUMOD events
- RTT alarm
- RTC alarm

74.9.1.2 Power Consumption

Backup mode configuration and measurements are defined as follows:

- RTC is running.
- Current measurement is as shown in the following figure.

Figure 74-49. Measurement on VBAT

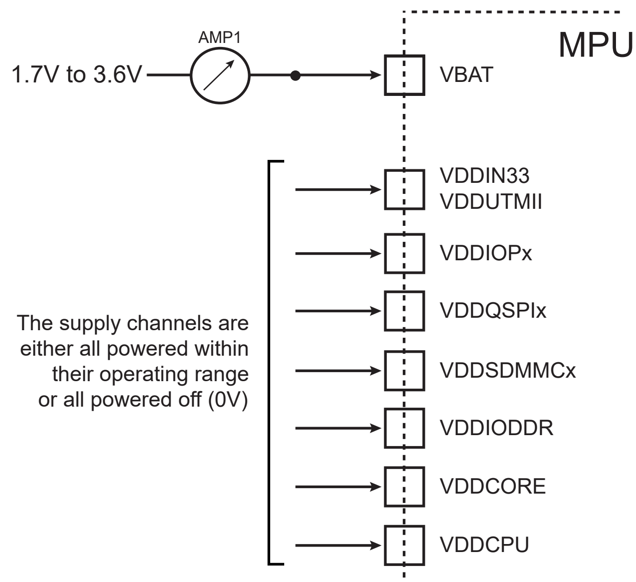


Table 74-71. Current Consumption on VBAT at T_J = 105°C

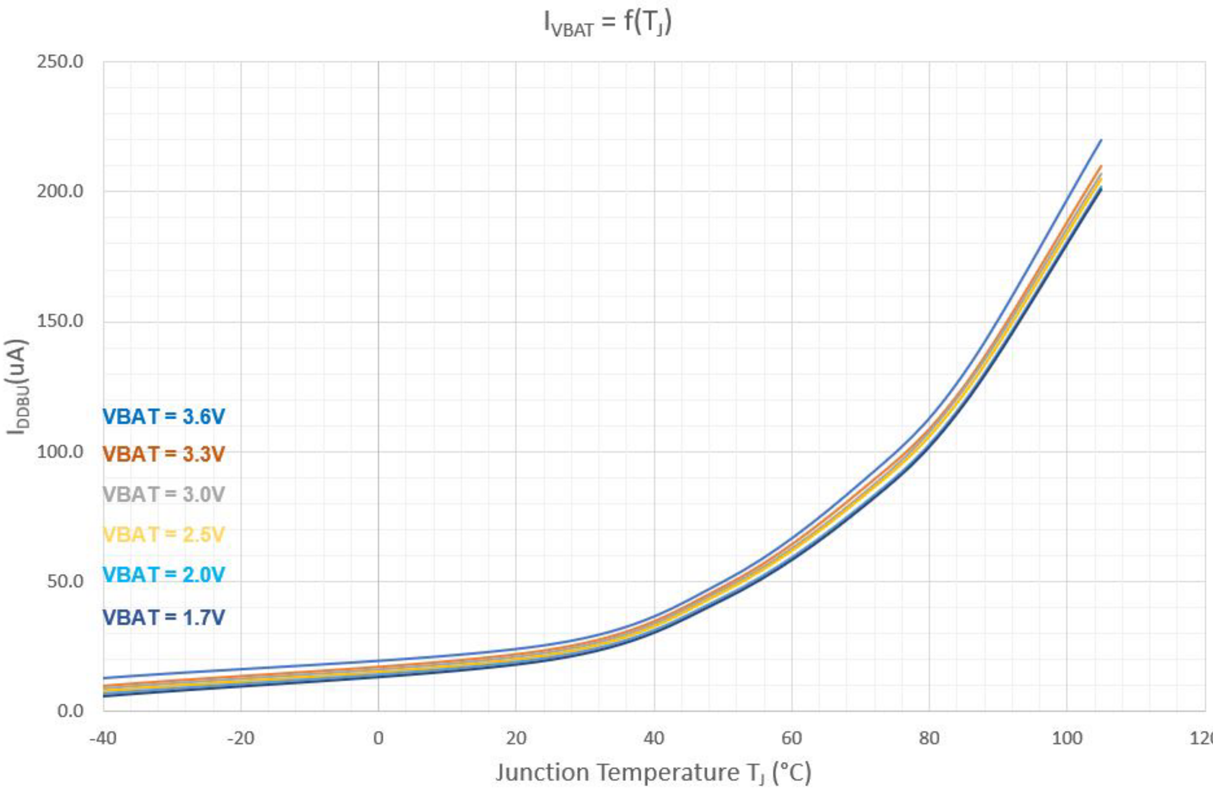
Symbol	Parameter	Conditions	Min	Max	Unit
I_{VBAT}	Current consumption on VBAT	VBAT = 3.6V - Backup mode[(1)]	–	1	mA
		VBAT = 3.6V - Normal mode[(2)(3)]	–	5	µA

Notes:

1. When operating the device in Backup mode at a lower junction temperature (70°C, for example), this current consumption specification can be derated using the VBAT current consumption characteristic given in the following figure.
2. The backup area embeds a VBAT-powered power switch which allows the VBAT section to be powered by VBAT (Backup mode) or by VDDIN33 when available (Normal mode). This power switch is controlled by bits CTRL and SOFTSWITCH in the SFRBU_PSWBU register. Refer to “SFRBU Power Switch BU Control Register” in the section Special Function Registers Backup (SFRBU) for more details.
3. Simulation data

The following figure shows the average Backup mode current consumption measured on a few typical devices.

Figure 74-50. Typical Current Consumption in Backup Mode



74.9.2 Backup with SDRAM in Self-Refresh (BSR) Mode

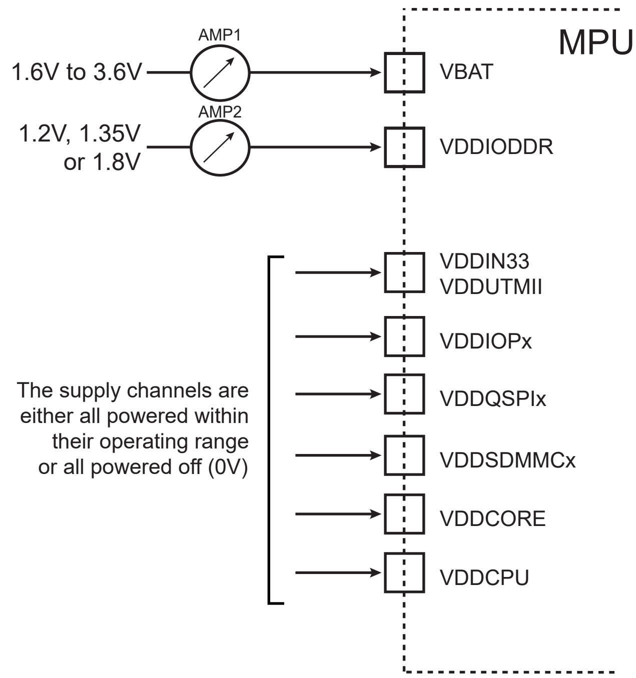
Backup with SDRAM in Self-Refresh (BSR) mode is an extension of Backup mode with the application context saved in the external DDR memory operating in Self-refresh mode, in the perspective to restart the application faster. In this mode, VBAT and VDDIODDR power inputs must be maintained, as well as the DDR component power inputs. It is good practice to use a combination of LPM and SHDN outputs from the Shutdown Controller (SHDWC) to signal the power management circuit to enter this special powering mode. See BSR Mode Entry and Wake-up for detailed waveforms.

The device exits BSR mode when an active wake-up event is triggered by the SHDWC. Upon this wake-up event, the SHDWC automatically toggles its SHDN output back to VBAT and this signal typically helps to restart all the power supply channels at board level. The LPM pin is not automatically reset by the SHDWC. It must be software-reset in the early steps of the wake-up process.

The system power consumption in BSR is mainly that of the DDR, therefore the choice of the DDR type is of prime importance. In a similar way, the choice of the regulator(s) that maintain(s) DDR supplies in BSR mode should optimize efficiency at low current.

74.9.2.1 Power Consumption in BSR Mode

Figure 74-51. Measurement on VBAT and VDDIODDR



Power consumption on AMP1 is provided in section Backup Mode. Only AMP2 measures are relevant in this section. The following table shows the result of average current measurements performed on a few typical devices.

Table 74-72. Typical Current Consumption in BSR Mode on VDDIODDR

VDDIODDR (V)	T_J=-40°C	T_J= 25°C	T_J= 50°C	T_J= 70°C	T_J=85°C	T_J=105°C	Unit
1.35V	< 1	2	5	15	30	75	µA

The following operations, based on procedures described in SDRAM Self-Refresh Mode, must be performed in this order. Code example is provided in software deliverables.

Entering Backup with SDRAM in Self-Refresh Mode

1. Enter Self-refresh mode with I/O power-down as described in the "Entering Self-Refresh Mode with I/O Power-down" procedure (see SDRAM Self-Refresh Mode).
2. Enter Backup mode:
   – Write a special code (0x12345678, for example) in a backup register to identify the BSR state.
   – Disable SYSC write protection.
   – Clear the SHDWC Control register.
   – Configure the wake-up source.
   – Send Shutdown and LPM Enable commands to the SHDWC Control register.

   Note: After wake-up, the device performs a system reset.
3. Re-initialize SDRAM and exit Self-refresh mode: refer to "High-Level SDRAM Initialization Procedure" in the section Universal DDR Memory Controller (UDDRC).

74.9.2.2 SDRAM Self-Refresh Mode

SDRAM devices require paced refreshes to maintain data consistency. These refreshes are paced by the DDR_CLK signal. To reduce system frequency or disable DDR_CLK, the SDRAM device must be put in Self-refresh mode.

Self-refresh deactivates the SDRAM clock and automatically executes a refresh operation using the SDRAM device internal refresh counter. Self-refresh mode is effective when the device is not accessed for a long time and the data must be kept for a future use (for context saving or peripheral configurations to recover current system state for example).

As the SDRAM device is not accessed, SDRAM I/Os are configured in Retention state:

- DDR_CLK tied to 0
- DDR_CLKN tied to 0
- DDR_CSN tied to 1

This is the case in some SAMA7G5 low-power modes such as Backup Self-refresh, ULP1 and ULP2. These modes use the following routines. Code example is provided in software deliverables.

Entering SDRAM Self-Refresh Mode

1. Save the first eight 32-bit SDRAM words in the backup SRAM.
2. Save the calibration result in the backup SRAM.
3. Set the static values for CK and CK# to 0,0 when the pair is disabled.
4. Perform a Data Synchronization Barrier (DSB).
5. Disable all UDDRC ports.
   – Wait for port disable to complete.
6. Move system to Self-refresh state.
   – Wait until Self-refresh is entered.
7. Put DDR3PHY DLLs in Bypass mode.
8. Power down the DDR3PHY data receivers.
9. Power down the Address Control, Clock, Chip Select and ODT pin output drivers.

Entering Self-Refresh Mode with I/O Power-down

1. Save the first eight 32-bit SDRAM words in the backup SRAM.
2. Save the calibration result in the backup SRAM.
3. Set the static values for CK and CK# to 0,0 when the pair is disabled.
4. Perform a Data Synchronization Barrier (DSB).
5. Disable all UDDRC ports.
   – Wait for port disable to complete.
6. Move system to Self-refresh state.
   – Wait until Self-refresh is entered.
7. Power down the DDR3PHY data receivers.
8. Power down the Clock and Chip Select pin output drivers.
9. Power down the Address Control and ODT pin output drivers.
10. Set the SDRAM I/Os to Retention state by setting SFRBU_DDRPWR.RETENTION.

Exiting SDRAM Self-Refresh Mode

1. Power down the CK and CS pin output drivers.
2. Power down the ODT[3:0] pin I/Os output drivers in the UDDRC DFI Low Power Configuration register 0 (UDDRC_DFILPCFG0).
3. Power down the data input receiver I/Os.
4. Move DDR3PHY DLLs out of Bypass mode.
5. In the UDDRC Software Register Programming Control Enable (UDDRC_SWCTL) register, enable quasi-dynamic register programming.
6. Clear the UDDRC_DFIMISC.DFI_INIT_COMPLETE_EN bit.
7. Set UDDRC_SWCTL.SW_DONE to indicate that UDDRC programming is completed.
   – Poll the UDDRC_SWSTAT.SW_DONE_ACK bit until it changes to 1, acknowledging that programming is complete.
8. Perform a DDR3PHY DLL soft reset , and DLL Lock and ITM soft reset commands.
   – Wait for completion.
9. In the UDDRC_SWCTL register, enable quasi-dynamic register programming.

10. Set the DDRC_DFIMISC.DFI_INIT_COMPLETE_EN bit.
11. Set UDDRC_SWCTL.SW_DONE to indicate that UDDRC programming is completed.
    – Poll the UDDRC_SWSTAT.SW_DONE_ACK bit until it changes to 1, acknowledging that programming is complete.
12. Trigger self-refresh exit.
    – Wait until Self-refresh state is exited.
13. Enable the five AXI ports with the UDDRC_PCTRL_x.PORT_EN bits.
14. Recover the first eight 32-bit SDRAM words from the backup SRAM.
15. Recover the calibration result from the backup SRAM.

74.9.3 Ultra Low-Power (ULP0, ULP1, ULP2) and Idle Modes

74.9.3.1 ULP0, ULP1 and Idle Modes

In ULP0, ULP1 and Idle low-power modes, all device power supplies are applied within their operating range. The power reduction is achieved by reducing the frequency or stopping the clock signals of the processor and/or its peripherals.

- In ULP0 or ULP1 mode, the device is able to resume on wake-up events (any interrupt or hardware event). This mode combines the Arm core Wait for Interrupt (WFI) mode and the system clocks frequency reduction or shut-off.
- In Idle mode, only the processor clock is affected; the clocks feeding the processor and its peripherals are slowed down or stopped.

A detailed description of each mode is provided in the following sections and is followed by a power consumption section dedicated to the modes.

74.9.3.1.1 Idle Mode Operation

In Idle mode, the device power consumption is optimized versus response time. In this mode, only the processor clock is stopped. The peripheral clocks, including the DDR controller clock, can be enabled. The current consumption in this mode is application-dependent.

This mode is entered via the WFI instruction and CPU_CLK disabling.

The processor can be awakened from an interrupt. The system resumes where it was before entering WFI mode.

74.9.3.1.2 ULP0 Mode Operation

ULP0 mode maintains very low frequency clocks (MCK1, CPU_CLK) in the system to wake up on any interrupt. Selection of the clock frequency depends on the current consumption target versus the required wake-up time. The higher the frequency, the higher the power consumption and the faster the wake-up time.

The sequence to enter ULP0 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM. Steps 1 to 6 are meant to bypass the DDR delay lines so as to avoid their power consumption.

Entering ULP0 Mode

1. Enter SDRAM Self-refresh mode (if used).
2. Turn on DDR PLL (if DDR was not used).
3. Reset the DDR Controller using RSTC_GRSTR.
4. Wait until the status bit is asserted.
5. Program the lane and control delay lines bypass using the DDLDIS bit in DDR3PHY_ACDLLCR, DDR3PHY_DX0DLLCR and DDR3PHY_DX1DLLCR registers.
6. Turn off DDRPLL in PMC.
7. Set the interrupts to wake up the system.
8. Disable all peripheral clocks.
9. Set the I/Os to an appropriate state and suspend the USB transceivers.
10. Switch the system clock to MD_SLCK.
11. Disable the PLLs, the Main Crystal Oscillator and the Main RC Oscillator.
12. Set Analog LDO to Low-power mode with the SFRBU_LDO25CR.LOWPOWER bit.
13. Enter the Wait for Interrupt (WFI) mode.

Exiting ULP0 Mode

Wake-up from ULP0 mode is triggered by any enabled interrupt. When resuming, the software reconfigures the system (oscillator, PLL, etc.) to the same state as before WFI. The Analog LDO must be set back to normal mode.

74.9.3.1.3 ULP1 Mode Operation

In ULP1 mode, all device power supplies are applied within their operating range. Power reduction is achieved by stopping the clock signals of the processor and/or its peripherals. The device is able to resume on wake-up events.

The following operations, based on procedures described in SDRAM Self-Refresh Mode, must be performed in this order. Code example is provided in software deliverables.

Entering ULP1 Mode

In ULP1 mode (unlike ULP0 mode) all clocks are off and the number of wake-up sources is limited to:

- Any PIO line configured as a wake source in PMC_WCR
- RTC alarm, RTT alarm
- USB Resume from Suspend mode

To enter ULP1 mode:

1. Enter SDRAM Self-refresh mode.
2. Clear all pending events.
3. Configure the wake-up source. RTT example:
   – Disable the interrupt for RTT.
   – Enable the RTT wake-up in the PMC Fast Start-Up Mode register (PMC_FSMR).
   – Set the RTT Alarm register (RTT_AR) to approximately 1 minute.
   – Perform an RTT Restart and enable the alarm interrupt.
4. Suspend USB ports 0, 1 and 2.
5. Disable all GCLK peripheral clocks.
6. Disable PMC protection.
7. Switch MCK0 to MAINCK.
8. Set MDIV to 1 for MCK0.
9. Switch the MCK source to MAINCK for all MCKs (1 to 4).
10. Set MDIV to 1 for all MCKs (1 to 4).
11. Turn off all eight PLLs.
12. Turn on the Main RC.
13. Switch MAINCK to Main RC.
14. Turn off the Main Crystal Oscillator.
15. Disable SYSC and SHDW write protections.
16. Configure the expected event (RTT alarm, for example).
17. Set Analog LDO to Low-power mode with the SFRBU_LDO25CR.LOWPOWER bit.
18. Set Ultra Low-power 1 mode in PMC.

The system is now in ULP1 mode, waiting for the programmed event (such as RTT alarm).

Exiting ULP1 Mode

When the event occurs, recover pre-ULP1 state as follows:

1. Start up the Main Crystal Oscillator.
2. Switch MAINCK to Main Crystal Oscillator.
3. Switch MCK0 to MAINCK.
4. Set MDIV to 1 for MCK0.
5. Start up CPUPLL.
6. Switch MCK0 to CPUPLL with MDIV set to 3.
7. Start up SYSPLL (400 MHz).
8. Switch MCK1 to SYSPLL with MDIV set to 2.

9. Switch MCK4 to SYSPLL with MDIV set to 1.
10. Start up DDRPLL (533 MHz).
11. Set Analog LDO out of Low-power mode.
12. Re-initialize SDRAM and exit Self-refresh mode.
13. Recover data from SDRAM.

74.9.3.2 ULP2 Mode Operation

In ULP2 mode, all device power supplies are applied within their operating range, except VDDCPU. The power reduction is achieved by stopping the clock signals of the processor and/or its peripherals. The device is able to resume on wake-up events. Additional power savings are made by shutting down the VDDCPU power supply and thus eliminating any leakage power in the Cortex-A7 subsystem.

In this mode, the CPU cluster (Cortex-A7 + L1 and L2 cache memories) is powered down while all other registers and memories of the device are maintained[(1)]. Upon wake-up triggered by an event (see detailed list), the VDDCPU power supply is re-applied and when the VDDCPU POR releases the CPU reset line, the device executes the ROM code @ 0x00000000.

Note:

1. Some content is overwritten due to ROM code execution.

A detailed description of each mode is provided in the following sections and is followed by a power consumption section dedicated to the modes.

The following operations, based on procedures described in SDRAM Self-Refresh Mode, must be performed in this order. Code example is provided in software deliverables.

Entering ULP2 Mode

In ULP2 mode (unlike ULP0 mode) all clocks are off and the number of wake-up sources is limited to:

- Any PIO line configured as a wake-up source in PMC_WCR
- RTC alarm, RTT alarm
- USB Resume from Suspend mode

To enter ULP2 mode:

1. Enter SDRAM Self-refresh mode.
2. Clear all pending events.
3. Configure the wake-up source. RTT example:
   – Disable the interrupt for RTT.
   – Enable the RTT wake-up in the PMC Fast Start-Up Mode register (PMC_FSMR).
   – Set the RTT Alarm register (RTT_AR) to approximately 1 minute.
   – Perform an RTT Restart and enable the alarm interrupt.
4. Suspend USB ports 0, 1 and 2.
5. Disable all GCLK peripheral clocks.
6. Disable PMC protection.
7. Switch MCK0 to MAINCK.
8. Set MDIV to 1 for MCK0.
9. Switch the MCK source to MAINCK for all MCKs (1 to 4).
10. Set MDIV to 1 for all MCKs (1 to 4).
11. Turn off all eight PLLs.
12. Turn on the Main RC.
13. Switch MAINCK to Main RC.
14. Turn off the Main Crystal Oscillator.
15. Disable SYSC and SHDW write protections.
16. Configure the expected event (RTT alarm, for example).
17. Configure LPM in Automatic mode.
18. Set Analog LDO to Low-power mode with the SFRBU_LDO25CR.LOWPOWER bit.

19. Set Ultra Low-power 2 mode in PMC.
20. Send a Wait for Event (WFE) command.

The system is now in ULP2 mode, waiting for the programmed event (such as RTT alarm).

Exiting ULP2 Mode

When the event occurs, the CPU power supply is recovered and the device executes the ROM code @ 0x00000000. The Reset Controller indicates "ULP Mode 2 reset" as reset type in the RSTC_SR.RSTTYP field. This flag is taken into account by the software to re-initialize the system and recover pre-ULP2 configuration.

A typical system re-initialization is as follows.

Regular ROM code initialization:

1. Switch MCK0 to MAINCK.
2. Set MDIV to 1 for MCK0.
3. Start up CPUPLL (570 MHz).
4. Switch MCK0 to CPUPLL with MDIV set to 4.
5. Start up SYSPLL (378 MHz).
6. Switch MCK1 to SYSPLL with MDIV set to 2.
7. Switch MCK4 to SYSPLL with MDIV set to 1.
8. Perform the required boot strategy.

User's configuration after exiting ULP2 (in user's code):

1. Start up the Main Crystal Oscillator.
2. Switch MAINCK to Main Crystal Oscillator.
3. Modify clocks configuration (MCKx, GCLKx, etc.).
4. Start up DDRPLL (533 MHz).
5. Set Analog LDO out of Low-power mode with the SFRBU_LDO25CR.LOWPOWER bit.
6. Re-initialize SDRAM and exit Self-refresh mode.
7. Recover data from SDRAM.

74.9.3.3 Power Consumption in Idle and Ultra Low-Power Modes

- Figure 74-53 and Figure 74-54 provide the SAMA7G5 average power consumption in Idle, ULP0, ULP1 and ULP2 modes versus temperature measured on a few typical devices. They do not provide maximum power consumption specifications.
- All power supply inputs are powered within their operating range and in particular VDDCORE = 1.15V, VDDCPU = 1.15V.
- There is no consumption on the device I/Os. To reach the best possible power consumption figures, it is important to set each I/O of the device to an appropriate state (pull-up/pull-down, etc.) with respect to the external components connected to these I/Os.
- USB tranceivers are disabled.
- All peripheral clocks are disabled.
- Current measurement is as shown in the following figure.

Figure 74-52. Current Measurement in ULP0, ULP1, ULP2 and Idle Modes

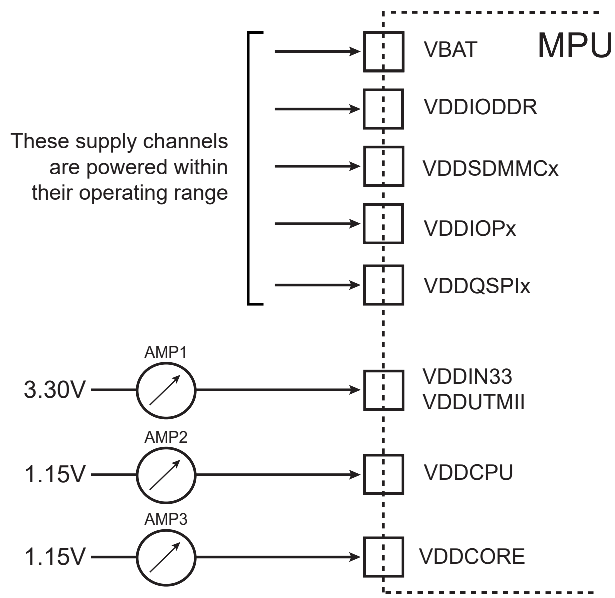


Table 74-73. Typical Current Consumption in Idle, ULP0, ULP1, ULP2 Modes

Low-Power Mode	IDDCPU / IDDCORE (mW)					
	T_J=-40°C	T_J= 25°C	T_J=50°C	T_J=70°C	T_J=85°C	T_J=105°C
Idle	30 / 115	42 / 123	56 / 130	75 / 140	95 / 151	132 / 172
ULP2	0 / 1	0 / 9	0 / 19	0 / 34	0 / 49	0 / 77
ULP1	1 / 1	13 / 9	27 / 19	46 / 34	66 / 49	103 / 77
ULP0 (12 MHz)	1 / 1	14 / 10	27 / 20	46 / 35	66 / 50	103 / 78
ULP0 (512 Hz)	1 / 1	13 / 9	27 / 19	46 / 34	66 / 46	103 / 77

The following figures plot the power consumption of the device in ULP0, ULP1, ULP2 and Idle modes. These figures account for the current consumption in VDDCPU, VDDCORE: Power (mW) = I_{DDCPU} x 1.15V + I_{DDCORE} x 1.15V with I_{DDCPU} and I_{DDCORE} as measured in the above table.

Figure 74-53. Typical Power Consumption in Idle Mode vs Junction Temperature

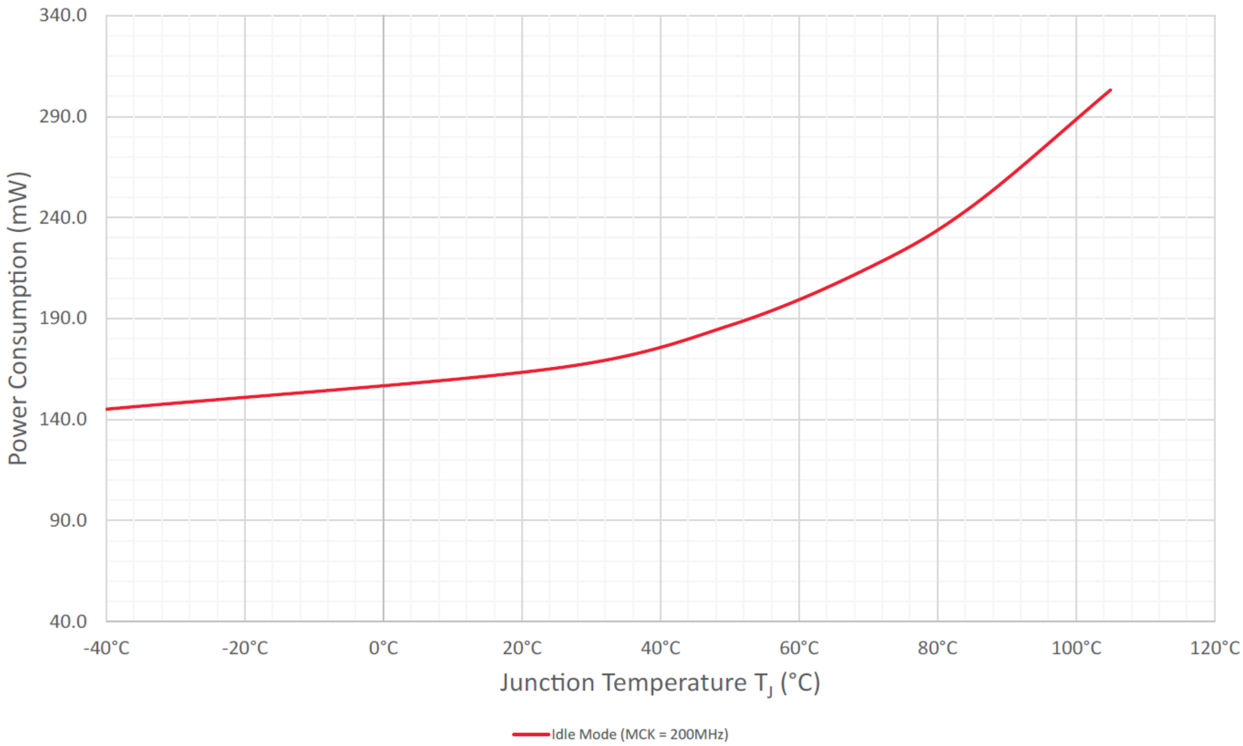
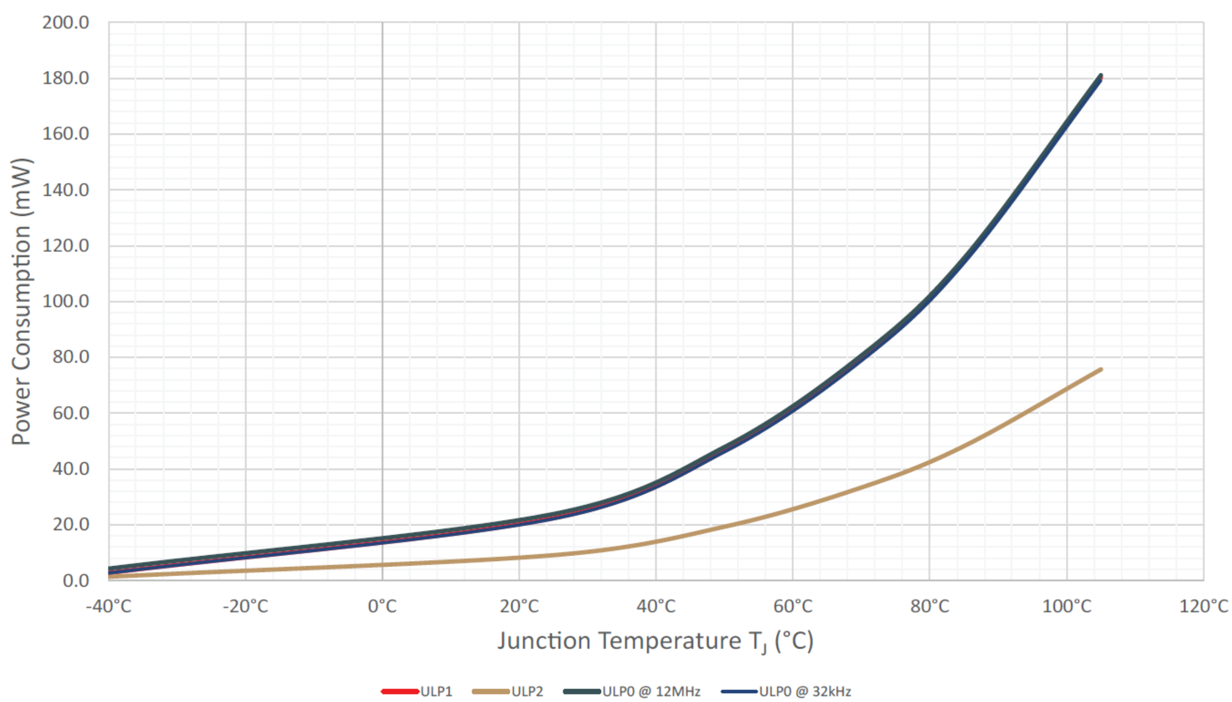


Figure 74-54. Typical Power Consumption in ULP0, ULP1 and ULP2 Modes vs Junction Temperature



The following table provides the specifications for ULP1 mode current consumption on VDDCORE, VDDCPU and VDDIN33.

Table 74-74. Power Consumption in ULP1 Mode at T_J=105°C

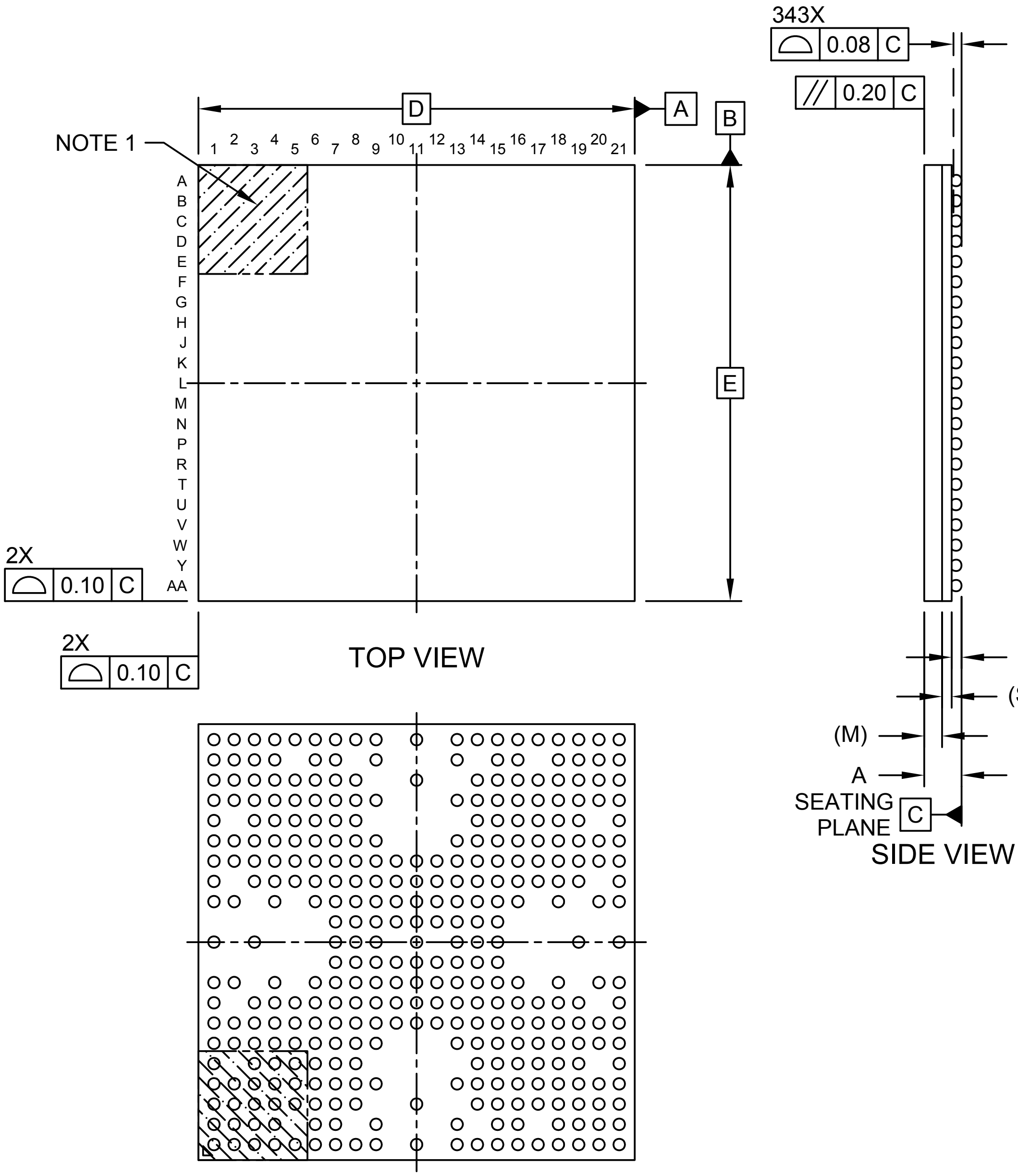
Symbol	Parameter	Conditions	Min	Max	Unit
P_{CORE_CPU}	Power consumption on VDDCORE and VDDCPU	VDDCORE = 1.15V VDDCPU = 1.25V	–	300	mW
P_{IN33}	Power consumption on VDDIN33	VDDIN33 = 3.6V	–	2.2	mW

75. Mechanical Characteristics

75.1 343-ball TFBGA Mechanical Characteristics

343-Ball Plastic Thin Fine PItch Ball Grid Array (4HB) - 14x14x1.2 mm Body [TFBGA]

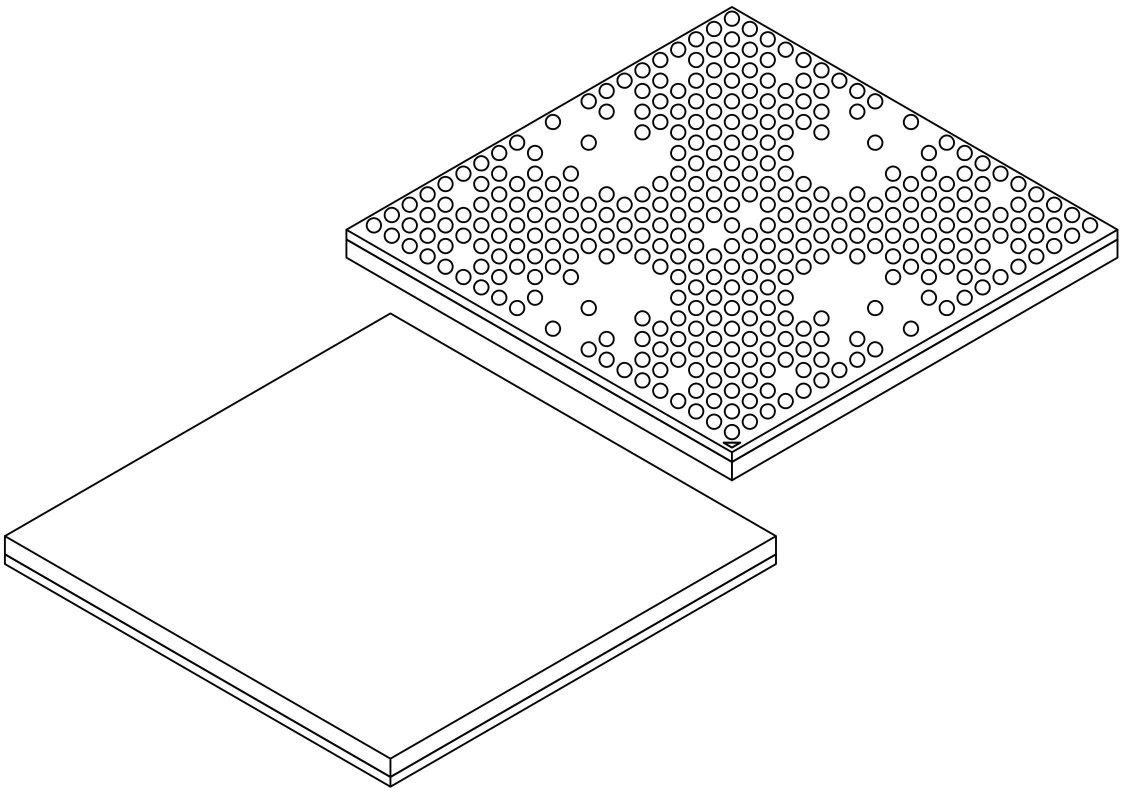
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-21516 Rev A Sheet 1 of 2

343-Ball Plastic Thin Fine PItch Ball Grid Array (4HB) - 14x14x1.2 mm Body [TFBGA]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Terminals	N	343		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	1.20
Standoff	A1	0.22	0.27	0.32
Substrate Thickness	S	0.26 REF		
Mold Cap Height	M	0.53 REF		
Overall Length	D	14.00 BSC		
Overall Width	E	14.00 BSC		
Ball Diameter	b	0.32	0.35	0.42

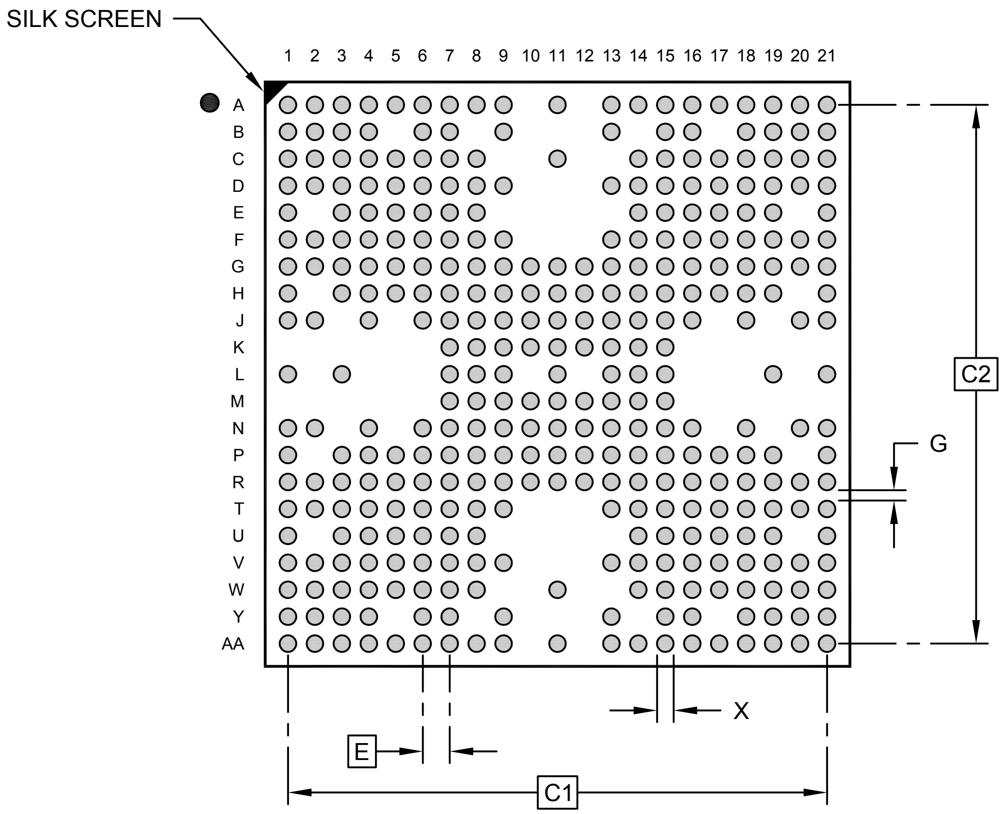
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21516 Rev A Sheet 2 of 2

343-Ball Plastic Thin Fine PItch Ball Grid Array (4HB) - 14x14x1.2 mm Body [TFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

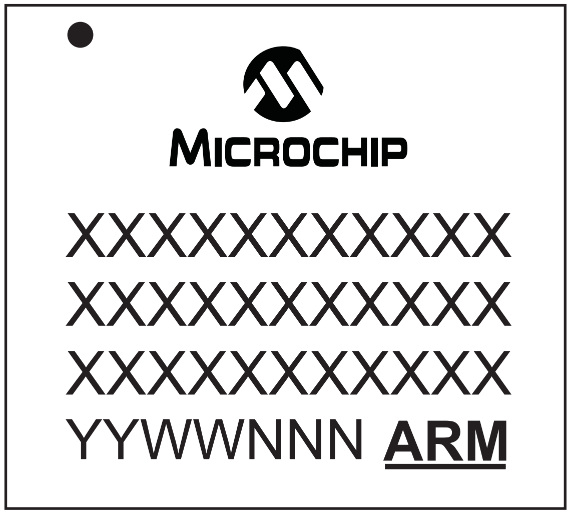
Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C1		13.00	
Contact Pad Spacing	C2		13.00	
Contact Pad Width (Xnn)	X			0.40
Contact Pad to Contact Pad (Xnn)	G	0.25		

Notes:
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-23516 Rev A

76. Marking

Top marking follows the scheme below:



with possible values:

Line	Description	Values
1	Company logo	Microchip logo
2	Company name	Microchip
3	Device name	SAMA7G54
4	Temperature code / Packaging code, Jedec symbol	V/4HB e3
5	Not used	–
6	Lot traceability, Arm logo	YYWWNNN ARM

77. Ordering Information

For details on ordering codes, refer to Product Identification System.

Ordering Code	Package	Carrier Type	Qualification Tier	Junction Temperature Range
SAMA7G54-V/4HB	TFBGA343	Tray	Industrial	-40°C to +105°C

78. Glossary

The following acronyms are used throughout this data sheet.

Table 78-1. Glossary

Acronym	Description
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard Controller
ASRC	Asynchronous Sample Rate Converter
CHIPID	Chip Identifier
DDRPUB	DDR SDRAM PHY Utility Block "Lite" aka PUBL
EBI	External Bus Interface
FLEXCOM	Flexible Serial Communication Controller (SPI+TWI+USART Controller)
GMAC	Gigabit Ethernet MAC
I2SMCC	Inter-IC Sound Multi-Channel Controller
ICM	Integrity Check Monitor
ISC	Image Sensor Controller
MATRIX	HSS AHB MATRIX
MCAN	Controller Area Network Controller
NICGPV	NIC-400 Global Programmer’s View (AXI MATRIX)
PDMC	Pulse Density Microphone Controller
PIO A-E	Peripheral PIO Controller
PMC	Power Management Controller
PWM	Pulse Width Modulation Controller
QSPI	Quad Serial Peripheral Interface
SDMMC	SD Card/MMC Controller
SECUMOD	Security Module
SECURAM	Secure Backup SRAM
SFR	Special Function Registers
SHA	Secure Hash Algorithm (SHA Encryption Controller)
SMC	Static Memory Controller
SPDIFRX	Sony Philips Digital Interface Receiver
SPDIFTX	Sony Philips Digital Interface Transmitter
SSC	Synchronous Serial Controller
SYSC	System Controller
TC	Timer Counter
TDES	Triple Data Encryption Standard
TRNG	True Random Number Generator

...........continued

Acronym	Description
TZAESB	TrustZone Advanced Encryption Standard Bridge
UDDRC	Universal DDR Memory Controller

79. Revision History

79.1 DS60001765A - 03/2022

Changes
First issue.

Table of Contents

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

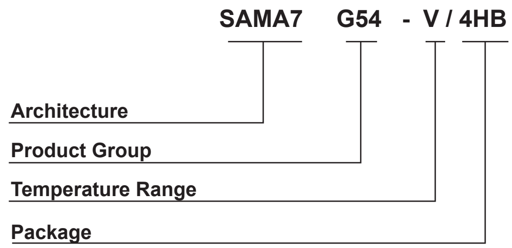
- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture:	SAMA7	= Arm Cortex-A7 CPU
Product Group:	GXX	= 343-ball general-purpose microprocessors with variable feature set
Temperature Range:	V	= -40°C to +105°C (industrial)
Package:	4HB	= 343-ball TFBGA

Description:

- SAMA7G54-V/4HB = Arm Cortex-A7 general-purpose microprocessor, industrial temperature, 343-ball TFBGA package

Note: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE

Trademarks

ISBN: 978-1-6683-0028-2

Quality Management System

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820